

# Monte Carlo Simulation of Polymers: Coarse-Grained Models

J. Baschnagel, J. P. Wittmer, and H. Meyer

Institut Charles Sadron,  
6, rue Boussingault, 67083 Strasbourg Cedex, France  
*E-mail:* {baschnag, jwittmer, hmeyer}@ics.u-strasbg.fr

A coarse-grained simulation model eliminates microscopic degrees of freedom and represents a polymer by a simplified structure. A priori, two classes of coarse-grained models may be distinguished: those which are designed for a specific polymer and reflect the underlying atomistic details to some extent, and those which retain only the most basic features of a polymer chain (chain connectivity, short-range excluded-volume interactions, etc.). In this article we mainly focus on the second class of generic polymer models, while the first class of specific coarse-grained models is only touched upon briefly. Generic models are suited to explore general and universal properties of polymer systems, which occur particularly in the limit of long chains. The simulation of long chains represents a challenging problem due to the large relaxation times involved. We present some of the Monte Carlo approaches contrived to cope with this problem. More specifically, our review contains two main parts. One part (Sec. 5) deals with local and non-local updates of a polymer. While local moves allow to extract information on the physical polymer dynamics from Monte Carlo simulations, the chief aim of non-local moves is to accelerate the relaxation of the polymers. We discuss some examples for such non-local moves: the slithering-snake algorithm, the pivot algorithm, and its recently suggested variant, the double-pivot algorithm, which is particularly suited for the simulations of concentrated polymer solutions or melts. The second part (Sec. 6) focuses on modern Monte Carlo methods that were inspired by the Rosenbluth-Rosenbluth algorithm proposed in the 1950s to simulate self-avoiding walks. The modern variants discussed comprise the configuration-bias Monte Carlo method, its recent extension, the recoil-growth algorithm, and the pruned-enriched Rosenbluth method, an algorithm particularly adapted to the simulation of attractively interacting polymers.

## 1 Introduction

Polymers are macromolecules in which  $N$  monomeric repeat units are connected to form long chains.<sup>a</sup> Experimentally the chain length  $N$  is large, typically  $10^3 \lesssim N \lesssim 10^5$ . The size of a chain ( $\sim 10^3 \text{Å}$ ) thus exceeds that of a monomer ( $\sim 1 \text{Å}$ ) by several orders of magnitude. However, contrary to granular materials,<sup>2</sup> the chain is not so large that thermal energy<sup>b</sup> would be unimportant. Not at all! Thermal energy is the important energy scale for polymers. It provokes conformational transitions so that the polymer can assume a multitude of different configurations at ambient conditions.<sup>c</sup>

<sup>a</sup>More precisely, this definition refers to “linear homopolymers”, i.e., linear chain molecules consisting of one monomer species only. By contrast, polymer chemistry can nowadays synthesize various other topologies, such as cyclic, star- or H-polymers. For a very commendable review on the physical chemistry of polymers see Ref. 1.

<sup>b</sup>Here, we mean the thermal energy supplied at ambient temperature, i.e.,  $k_B T = 4.1 \cdot 10^{-21} \text{ J}$  for  $T = 300 \text{ K}$ .  
<sup>c</sup>Polymers are a paradigm for “soft matter” materials or “complex fluids”. Roughly speaking, “soft matter” consists of materials whose constituents have a mesoscopic size (microscopic scale  $\sim 1 \text{Å} \ll$  mesoscopic object  $\sim 10^2 - 10^4 \text{Å} \ll$  macroscopic scale  $\sim 1 \text{mm}$ ) and for which  $k_B T$  is the important energy scale (whence the softness at ambient conditions). Examples other than polymers are colloidal suspensions, liquid crystals, or fluid membranes.<sup>3</sup>

Changes of the configurations occur on very different scales, ranging from the local scale of a bond to the global scale of the chain.<sup>4,5</sup> This separation of length scales entails simplifications and difficulties. Simplifications arise on large scales where the chain exhibits universal behavior. That is, properties which are independent of chemical details.<sup>6,7</sup> These properties may be studied by simplified, “coarse-grained” models, e.g. via computer simulations. For simulations the large-scale properties, however, also give rise to a principal difficulty. Long relaxation times are associated with large chain lengths.<sup>7-9</sup>

The present chapter focuses on some of the Monte Carlo approaches to cope with this difficulty. Why Monte Carlo? Within a computational framework it appears natural to address dynamical problems via the techniques of Molecular Dynamics (see Ref. 10). A Molecular Dynamics (MD) simulation numerically integrates the equations of motion of the (polymer) system, and thereby replicates, authentically, its (classical) dynamics. As the polymer dynamics ranges from the (fast) local motion of the monomers to (slow) large-scale rearrangements of a chain, there is a large spread in time scales. The authenticity of MD thus carries a price: Efficient equilibration and sampling of equilibrium properties becomes very tedious –sometimes even impossible– for long chains. At that point, Monte Carlo simulations may provide an alternative. Monte Carlo moves are not bound to be local. They can be tailored to alter large portions of a chain, thereby promising efficient equilibration. The discussion of such moves is one of the gists of this review.

*Outline and Prerequisites.* The plan of the chapter is as follows: We begin by gathering necessary background information, both as to polymer physics (Sec. 2) and as to the Monte Carlo method (Sec. 3). Then, we present the simulation models (Sec. 4), which have been used to develop and to study various Monte Carlo algorithms. The discussion of these algorithms (Secs. 5 and 6) represents the core of the chapter. Section 5 deals with local moves, allowing to study the physical polymer dynamics via Monte Carlo, and non-local moves (slithering-snake algorithm, pivot algorithm, double-pivot algorithm), aiming at speeding up the relaxation of the chains. Section 6 discusses the Rosenbluth-Rosenbluth method for simulating self-avoiding walks and some of its modern variants (pruned-enriched Rosenbluth method, configuration-bias Monte Carlo, recoil-growth algorithm). The last section (Sec. 7) briefly recapitulates the different methods and gives some advice when to employ which algorithm. Finally, the appendix A reviews a recently proposed approach to systematically derive coarse-grained models for specific polymers.

Our presentation is based on the following prerequisites:

- We will restrict our attention to homopolymers, i.e., to polymers consisting of one monomer species only. However, (some of) the algorithms discussed may also be applied e.g. to polymer blends or block-copolymers (see Ref. 11).
- The chains are monodisperse, i.e.  $N$  is constant.
- We do not consider long-range (e.g., electrostatic) or specific (e.g., H-bonds) interactions between the monomers. These interactions are treated in other chapters (e.g., see Refs. 12, 13).
- We do not treat the solvent molecules explicitly. They are indirectly accounted for by the interactions between the monomers. The neglect of the solvent does not affect

the static properties of chains in dilute solution. However, it does affect their physical dynamics (see Ref. 14).

## 2 A Primer to Polymer Physics

### 2.1 A Polymer in Good Solvent

To substantiate the remarks of the introduction about the large-scale properties of polymers let us consider a specific example, a dilute solution of polyethylene. Polyethylene consists of  $\text{CH}_2$ -monomers which are joined to form a linear polymer (Fig. 1). A configuration of the chain may be specified by the positions of the monomers<sup>d</sup>  $\mathbf{x} = (\vec{r}_1, \dots, \vec{r}_N)$ . Thermodynamic properties are calculated by averaging an observable  $A$  over all configurations

$$\langle A \rangle = \frac{1}{\mathcal{Z}} \int d\mathbf{x} A(\mathbf{x}) \exp [ - \beta U(\mathbf{x}) ] . \quad (1)$$

Here  $\beta = k_B T$ ,  $\mathcal{Z}$  is the partition function and  $U(\mathbf{x})$  the interaction potential. We assume that  $U(\mathbf{x})$  can be split into two parts:<sup>e</sup>

$$U(\mathbf{x}) = \sum_{i=1}^{N-1} \underbrace{U_0(\vec{b}_i, \dots, \vec{b}_j, \dots, \vec{b}_{i+i_{\max}})}_{\text{"short-range": } \ell, \theta, \phi, \dots} + \underbrace{U_1(\mathbf{x}, \text{solvent})}_{\text{"long-range"}}, \quad (2)$$

where  $\vec{b}_i = \vec{r}_{i+1} - \vec{r}_i$  denotes the bond vector from the  $i$ th to the  $(i+1)$ th monomer.

The first term of Eq. (2),  $U_0$ , depends on the chemical nature of the polymer. It comprises the potentials of the bond length  $\ell$ , the bond angle  $\theta$ , the torsional angle  $\phi$ , etc. (Fig. 1).<sup>17</sup> These potentials lead to correlations between the bond vectors  $\vec{b}_i$  and  $\vec{b}_j$ . Typically, the correlations are of short range: they only extend up to some bond vector  $b_{j=i+i_{\max}}$  with  $i_{\max} \ll N$ .

Although distant monomers along the backbone of the chain are thus orientationally decorrelated, they can still come close in space. The resulting interaction is long-range along the chain backbone (Fig. 1). In Eq. (2), it is accounted for by the second term  $U_1$ .<sup>6,7</sup>  $U_1$  depends strongly on the quality of the solvent.<sup>f</sup> In good solvents the monomers effectively repel one another (they want to be surrounded by solvent molecules), whereas they attract each other if the solvent cannot dissolve the polymer (bad solvent).

Due to its long-range character, one intuitively expects  $U_1$  to influence the large-scale behavior of the chain more strongly than  $U_0$ . A possible test of this idea is to estimate how the size of a chain scales with  $N$ . Common measures of the chain size are the mean-square end-to-end distance  $R_e^2$  or the radius of gyration  $R_g^2$  (Fig. 1)

$$R_e^2 = \langle (\vec{r}_N - \vec{r}_1)^2 \rangle, \quad R_g^2 = \frac{1}{N} \sum_{i=1}^N \langle (\vec{r}_i - \vec{R}_{\text{cm}})^2 \rangle, \quad (3)$$

<sup>d</sup>Here, we adopt a description in terms of a so-called “united atom model”. The united atom model represents a  $\text{CH}_2$ -group by a single, spherical interaction site and does not distinguish between inner ( $\text{CH}_2$ ) and end monomers ( $\text{CH}_3$ ).<sup>15</sup> Furthermore, we neglect the momenta of the monomers to specify the configuration, as we assume the observables and interaction potentials to depend on positions only.

<sup>e</sup>Equation (1) does not contain the degrees of freedom of the solvent. They are assumed to be integrated out. Thus,  $U(\mathbf{x})$  is an effective potential—in fact, a free energy—depending on the properties of the solvent.

<sup>f</sup>In Eq. (2) we assume that  $U_0$  is independent of the solvent quality.

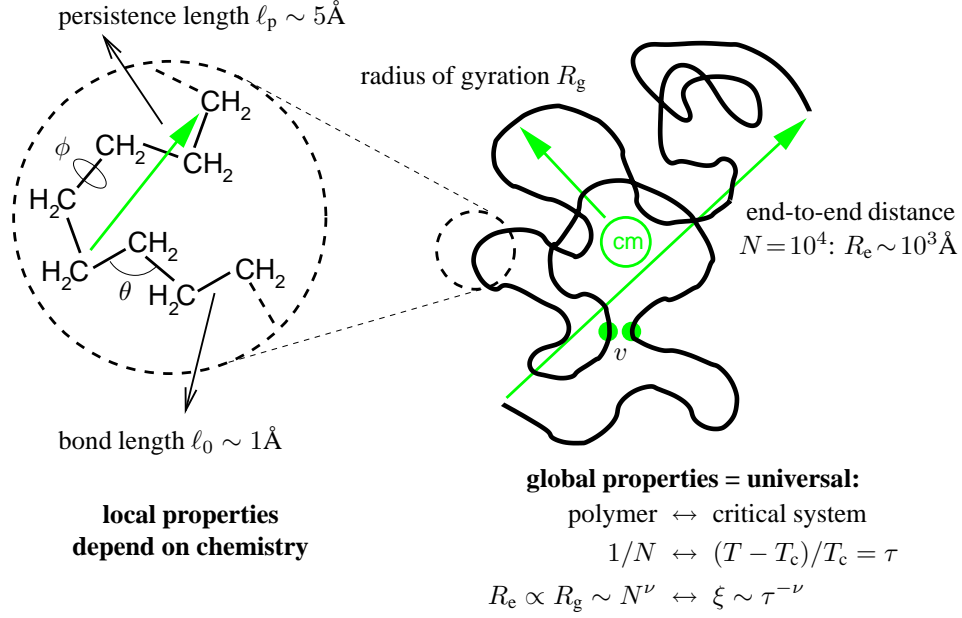


Figure 1. Schematic illustration of polyethylene. The local properties of the polymer depend on its microscopic degrees of freedom: the bond length  $\ell$ , the bond angle  $\theta$ , and the torsional angle  $\phi$ . Because the potential of the bond length is fairly “stiff”,  $\ell$  may be kept fixed at its equilibrium value  $\ell_0$  in a modeling approach. By contrast, the potential of the torsional angle is much “softer”. Thus,  $\phi$ , which characterizes rotations about a middle C-C bond, mainly determines the local conformation of the chain. All degrees of freedom ( $\ell, \theta, \phi$ ) determine the intrinsic stiffness of the chain. The stiffness reflects the persistence of orientational correlations along the backbone of the chain. Orientational correlations decouple on the length scale of the “persistence length”  $\ell_p$ . For typical chain lengths,  $N \sim 10^4$ ,  $\ell_p$  is much smaller than the end-to-end distance  $R_e$  or the radius of gyration  $R_g$ . ( $R_g$  measures the average distance of a monomer from the center of mass (cm) of the chain.) Thus, the chain appears flexible on length scales larger than  $\ell_p$ . If the polymer is dissolved in a good solvent, distant monomers (filled grey circles) repel each other when they come in contact. That is, the excluded-volume parameter  $v$ , measuring the effective interaction between distant monomers along the chain, is positive. Under these conditions (i.e., linear polymer with some flexibility and repulsive monomer-monomer interactions) a correspondence between the large-scale properties of the polymer and a critical system close to its phase transition can be established:<sup>6,16</sup>  $1/N$  may be identified with the reduced distance,  $\tau$ , to the critical temperature  $T_c$  of the phase transition, and  $R_e$  or  $R_g$  scale with  $N$  as the correlation length  $\xi$  of the order parameter does with  $\tau$ .  $\nu$  is a universal critical exponent, often called “Flory exponent” in polymer science.

where  $\vec{R}_{\text{cm}}$  is the position of the chain’s center of mass. Because  $R_e \propto R_g$  we focus on  $R_e$  in the sequel to illustrate the role played by  $U_0$  and  $U_1$ .

Let  $\hat{b}_i$  denote the unit vector associated with the bond  $\vec{b}_i$  of fixed length  $\ell_0$  (Fig. 1). Then, quite generally, we may write  $R_e^2$  as

$$R_e^2 = \ell_0^2 \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \langle \hat{b}_i \cdot \hat{b}_j \rangle = 2\ell_0^2 \sum_{i=1}^{N-1} \sum_{k=0}^{N-1-i} \langle \hat{b}_i \cdot \hat{b}_{i+k} \rangle - (N-1)\ell_0^2. \quad (4)$$

Apparently, the large-scale behavior of  $R_e$  depends on the range of orientational correlations between bond vectors. Two cases may be distinguished:<sup>9</sup>

<sup>9</sup>In part, the subsequent discussion closely follows that on p. 148 of Ref. 18.

1. If  $\langle \hat{b}_i \cdot \hat{b}_{i+k} \rangle$  is “short-range”, i.e., if it decays more rapidly than  $1/k$  for large  $k$ , the second term converges in the large- $N$  limit. Then,

$$R_e^2 = N\ell_0^2 \left[ 2 \sum_{k=0}^{\infty} \langle \hat{b}_1 \cdot \hat{b}_{1+k} \rangle - 1 \right] =: N\ell_0^2 \left[ 2 \frac{\ell_p}{\ell_0} - 1 \right] \quad (N \rightarrow \infty), \quad (5)$$

where we introduce the persistence length  $\ell_p$  in the last term. ( $\ell_p$  measures the “persistence” of orientational correlations along the backbone and thus the intrinsic stiffness of the chain; see Fig. 1). Equation (5) shows that short-range orientational correlations only affect the prefactor—they renormalize the bond length to  $b = \ell_0[2(\ell_p/\ell_0) - 1]^{1/2}$  ( $b$  is called “effective bond length”<sup>7</sup>)—but they do not change the scaling of  $R_e$  with  $N$ . The scaling is always “random-walk-like”:<sup>h</sup>  $R_e \sim N^{1/2}$ .<sup>7</sup> In polymer science, a chain exhibiting this random-walk-like behavior is commonly referred to as an “ideal chain”.

Of course, the finite-range correlations, assumed for  $U_0$  in Eq. (2), are also of short range. Thus, provided  $U_1 = 0$ , the end-to-end distance of a (long) chain is given by  $R_e = bN^{1/2}$ , irrespective of the precise form of  $U_0$ . The microscopic degrees of freedom,  $\ell, \theta, \phi$ , determine the prefactor, the effective bond length  $b$ , but not the scaling with  $N$ . Therefore, if we are interested in studying large-scale properties, we can replace a chemically realistic model for polyethylene by a much simpler “coarse-grained model”, which is microscopically unrealistic, but correctly reproduces the large- $N$  behavior. An example for such a coarse-grained model is a “bead-spring model”, where  $N$  effective monomers (“beads”) are connected by harmonic springs of average length  $b$  (Fig. 2).

2. However, if  $\langle \hat{b}_i \cdot \hat{b}_{i+k} \rangle$  decays as  $1/k$  or more slowly (as  $1/k^y$  with  $y < 1$ ) due to long-range correlations, the scaling behavior of  $R_e^2$  is changed. Instead of  $R_e^2 \sim N$  we find

$$R_e^2 \sim N\ell_0^2 \int^N dk \langle \hat{b}(k) \cdot \hat{b}(0) \rangle \sim \begin{cases} N^{2-y} & (y < 1), \\ N \ln N & (y = 1). \end{cases} \quad (6)$$

Thus, long-range correlations lead to a “swelling” of the chain size with respect to a pure random walk.

Such long-range correlations are embodied in the potential  $U_1$  in Eq. (2). For a polymer in a good solvent a swelling of the chain dimension is physically reasonable. As soon as two (distant) monomers come close in space, they repel each other. On the level of the coarse-grained bead-spring model we can incorporate this repulsive interaction by writing  $U_1$  as (see e.g. Ref. 7 or the lucid discussion on pp. 16–20 of Ref. 16)

$$U_1(\vec{r}^N) = \int d^3\vec{r} \frac{1}{2} k_B T v \rho(\vec{r})^2 \quad \text{with} \quad \rho(\vec{r}) = \sum_{i=1}^N \delta(\vec{r} - \vec{r}_i). \quad (7)$$

<sup>h</sup>By the term “random-walk-like” we mean the diffusional motion of a Brownian particle. This motion can be thought of as resulting from the addition of many small displacements in random directions so that the overall mean-square displacement of the particle in time  $t$ ,  $R^2(t) = \langle [\vec{r}(t) - \vec{r}(0)]^2 \rangle$ , scales as  $R^2 \sim t$ . This allows for the following identifications in regard to polymer physics:  $R \leftrightarrow R_e$  and  $t \leftrightarrow N$ .

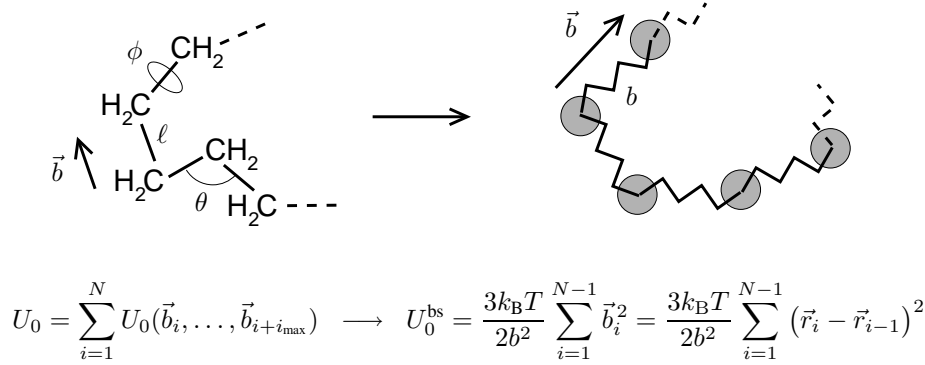


Figure 2. From a chemically realistic model to a coarse-grained bead-spring model. Local properties of the realistic model are determined by its microscopic degrees of freedom:  $\ell$ ,  $\theta$ , and  $\phi$ . On the global level of the chain, however, the influence of the microscopic degrees of freedom can be lumped into one parameter, the effective bond length  $b$ . The microscopic degrees of freedom do not determine the scaling of the end-to-end distance, which, under the sole effect of  $U_0$ , is given by  $R_e = bN^{1/2}$  (“ideal chain”). This behavior may be recovered from Eq. (1) when calculating  $R_e$  with the potential  $U_0^{\text{bs}}$  of a coarse-grained bead-spring model. This model identifies the monomers with spherical “beads” which are bound to one another by harmonic springs with force constant  $3k_B T/b^2$ . (This bead-spring model is often called “Gaussian chain” model in the polymer literature.<sup>7</sup>)

Here,  $\rho(\vec{r})$  is the monomer density at point  $\vec{r}$  and  $v (> 0)$  is the excluded-volume parameter.  $v$  measures the strength of the repulsion of a binary contact between two beads. Because a binary contact occurs with probability  $\rho(\vec{r})^2$ , Eq. (7) expresses the total energy penalty resulting from the repulsive contacts of all beads in the chain.

From the previous discussion of  $U_0$  and  $U_1$  the following conclusion may be drawn: When focusing on the large-scale properties of linear polymers with some flexibility and predominantly repulsive interactions we may forego a microscopic description in favor of a coarse-grained model. An example is the bead-spring model introduced above (Fig. 2), which is characterized by two parameters,  $b$  and  $v$ . Another possibility is a self-avoiding walk (SAW) on a (hyper-cubic) lattice. That is, a random walk which is not allowed to visit an already occupied lattice site again (see Sec. 4.1). The replacement “microscopic model  $\rightarrow$  SAW” is permissible because a linear polymer in good solvent can be shown to correspond to a critical system which undergoes a phase transition for  $N \rightarrow \infty$  (Fig. 1). It belongs to the universality class of the  $n$ -vector model in the limit  $n \rightarrow 0$ .<sup>6,16</sup> This implies that the large- $N$  behavior is determined by critical exponents. For instance,

$$R_g \propto R_e = bN^\nu \quad \text{or} \quad \mathcal{Z} \sim \mu^N N^{\gamma-1} \quad (N \rightarrow \infty), \quad (8)$$

where the partition function  $\mathcal{Z}$  counts the number of  $N$ -step SAW’s starting at the origin and ending anywhere. The connectivity constant  $\mu$  and the bond length  $b$  are non-universal. They depend on the polymer and the external conditions (temperature, solvent, etc.). By contrast, the critical exponents  $\nu$  and  $\gamma$  are universal. They only depend on the dimension of space.<sup>i</sup> Thus, they can be determined for all polymers by studying this (simple) model.

<sup>i</sup>In the course of the research on critical phenomena it has become clear that all systems with short-range, isotropic

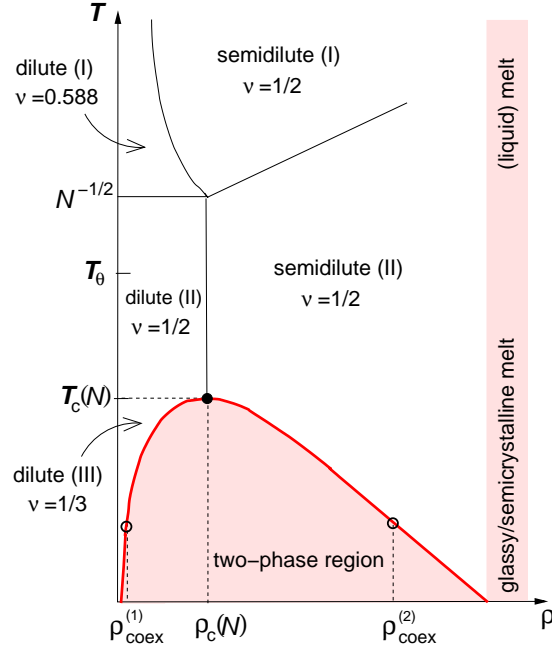


Figure 3. Schematic phase diagram of flexible polymers (see Chap. 9 of Ref. 16 or Chap. 4 of Ref. 23). For small monomer density  $\rho$  the solution is dilute. Three different regimes may be distinguished according to the temperature  $T$ : swollen chains [Eq. (8),  $T > T_\Theta$ : dilute (I)], nearly ideal chains [Eq. (10),  $T \approx T_\Theta$ : dilute (II)], and collapsed chains [Eq. (11),  $T < T_\Theta$ : dilute (III)]. There is an interval  $\Delta T$  around the  $\Theta$ -point of order  $\Delta T/T_\Theta \sim 1/\sqrt{N}$ , where the chains are nearly ideal. Whereas the chains may be considered as being isolated in dilute solution, they strongly overlap in the semidilute regimes. For  $T \leq T_c(N)$  phase separation in a dilute phase of collapsed chains and a semidilute solution of nearly ideal chains occurs. If the monomer density approaches 1, we obtain a polymer melt. At high  $T$  the melt is a (viscous) liquid, whereas at low  $T$  it may become a glassy<sup>24</sup> or a semicrystalline<sup>25</sup> solid, depending on the ability of the polymer to form ordered structures or not.

In fact, the currently most precise values of  $\nu$  and  $\gamma$  (see footnote on page 30) have been obtained from high-precision Monte Carlo simulations of SAW's.<sup>21,22</sup>

## 2.2 Phase Diagram of a Polymer Solution

The utility of coarse-grained models to investigate the statistical physics of polymer systems is not limited to the previous example. A dilute solution in a good solvent is just one region in the phase diagram. The phase diagram of flexible polymers is schematically shown in Fig. 3. Out of the various regimes we choose to discuss two cases in more detail, a chain in another than good solvent and (high-temperature) polymer melts. In the following sections we concentrate on those cases because novel Monte Carlo approaches have been applied to them.

interactions, the same dimension of space  $d$ , and the same dimensionality  $n$  of the order parameter ( $n = 1$ : scalar,  $n \geq 2$ :  $n$ -dimensional vector) have critical exponents which depend only on  $(d, n)$  and take the same values as those of the  $n$ -vector model.<sup>19,20</sup>

*A Chain in a  $\Theta$ -Solvent or a Bad Solvent.* To extend the discussion of the good solvent to other solvents let us reconsider Eq. (7). This equation corresponds to the first term of a virial expansion in the monomer density  $\rho(\vec{r})$ . That is,<sup>7,16</sup>

$$U_1 = \int d^3\vec{r} \left[ \frac{1}{2} k_B T v \rho(\vec{r})^2 + \frac{1}{6} k_B T w \rho(\vec{r})^3 + \dots \right]. \quad (9)$$

This identifies the excluded-volume parameter  $v$  with the second virial coefficient. In general, the virial coefficients depend on temperature  $T$ . The second virial coefficient vanishes at some temperature, called “ $\Theta$ -temperature  $T_\Theta$ ” in polymer science, and behaves as  $v = v_0(1 - T_\Theta/T)$  close to the  $\Theta$ -point ( $v_0 = \text{const.} > 0$ ). This implies that we can tune the solvent quality by temperature. In addition to the case of a good solvent ( $T > T_\Theta$ ) two further cases may be distinguished:

1.  *$\Theta$ -solvent* ( $T = T_\Theta$ ): Since binary interactions are absent [but ternary interactions are present:  $w > 0$  in Eq. (9)], the polymer behaves nearly as an ideal chain:<sup>16</sup>

$$R_e \propto R_g \sim \sqrt{N} \quad (+ \ln N \text{ corrections}) . \quad (10)$$

2. *Bad solvent* ( $T < T_\Theta$ ): Since the binary interactions are attractive, the polymer is collapsed to a dense sphere of monomers, implying that the average monomer density  $\rho$  inside the sphere is of order 1. Thus,

$$\rho \approx \frac{N}{R_g^3} \sim 1 \quad \Rightarrow \quad R_e \propto R_g \sim N^\nu \quad \text{with} \quad \nu = \frac{1}{3} . \quad (11)$$

The simulation of this situation is complicated because the time to equilibrate the chain and to sample equilibrium properties from many independent configurations becomes exceedingly long. Two factors are responsible for that. On the one hand, the local dynamics is sluggish (maybe even glass-like) due to the dense packing of monomers that strongly attract each other. On the other hand, the polymer is entangled with itself. Bonds cannot pass through each other. These topological constraints may also lead to slow dynamics for long chains.

*The Size of a Chain in a Polymer Melt.* In a good solvent a chain expands with respect to the ideal state, owing to long-range monomer-monomer repulsions. This is peculiar to dilute solutions. In a dense liquid of chains, a “polymer melt”, the situation is quite different. One can show<sup>6,7,26</sup> that the intra-chain excluded-volume interactions are screened by the presence of the surrounding polymers. Thus, a chain in a melt behaves on large scales as an ideal chain, implying  $R_e \propto R_g \sim N^{1/2}$  (see Fig. 4).

This ideality, first proposed by Flory,<sup>17</sup> appears fairly unexpected. Some feeling why this should be so may be obtained from the following argument: Inside the volume of a chain ( $\sim R_g^3$ ) the monomer density resulting from the  $N$  monomers of the chain is very small. For ideal chains it is of order  $N/R_g^3 \sim N^{-1/2}$ , whereas it scales as  $\sim N^{-0.764}$  under good solvent conditions (using Eq. (8) and  $\nu = 0.588$ ). We see that in dilute solution, swelling reduces the monomer density inside the chain and thus the total interaction energy [see Eq. (7)]. However, no energetic advantage may be gained in a melt because the overall monomer density is  $\rho \sim 1$ . Swelling would reduce the number of intra-chain contacts, but this reduction must be compensated by inter-chain contacts to keep  $\rho$  constant. Thus, a chain has to have  $N^{1/2}$  contacts with other chains, which is huge in the large- $N$  limit. This strong interpenetration of the chains suppresses the expansion of an individual chain.



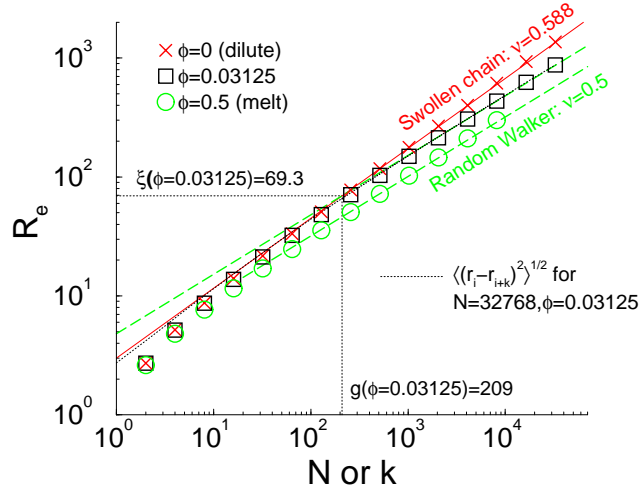


Figure 4. End-to-end distance  $R_e$  versus chain length  $N$  for the (athermal) bond-fluctuation model which will be discussed in more detail in Secs. 4.1 and Sec. 5. Results for three volume fractions (of occupied lattice sites) are given, illustrating the dilute ( $\phi = 0$ ), the semidilute ( $\phi = 0.03125$ ) and the melt ( $\phi = 0.5$ ) limits of the schematic phase diagram (Fig. 3). Using the slithering-snake algorithm (Sec. 5.2) it is possible to simulate chains containing up to  $N = 32768$  monomers for  $\phi < 0.5$ . Since the slithering-snake algorithm becomes less efficient at high densities (Sec. 5.2), the recently proposed double-pivot algorithm, described in Sec. 5.3, was harnessed to probe systems of higher densities ( $\phi = 0.5$ ). Periodic boxes of linear size  $L = 512$  and containing up to  $2^{22}$  monomers are required to eliminate finite-size effects. Such periodic boundary conditions are not needed for single chains. Here, an infinite box was used ( $L(\phi = 0) = \infty$ ). As only excluded-volume interactions are taken into account, good solvent statistics applies in dilute solution. The chains are swollen, as indicated by the exponent  $\nu = 0.588$  (solid line), which fits the data over three orders of magnitude. In the opposite (so-called) melt limit long-range correlations appear to be screened down to small chain lengths of about  $N \approx 10$  (grey dashed line).<sup>27</sup> Both chain statistics are visible for the intermediate density ( $\phi = 0.03125$ ): Small chains ( $N \ll g$ ,  $R_e \ll \xi$ ) are swollen (solid line) and long chains are Gaussian (dashed line). The intercept of both lines defines the size  $\xi$  of the “excluded volume blob”<sup>6,7</sup> and the number of monomers  $g$  that the blob contains. The indicated numbers are specific to the volume fraction (and persistence length) given, but are independent of chain length. For a given density  $\xi$  corresponds to the chain size where the coils start to overlap. Also presented in the figure is the spatial distance  $\langle (\vec{r}_i - \vec{r}_{i+k})^2 \rangle^{1/2}$  along the longest chain for  $\phi = 0.03125$  (dotted line). With the exception of small  $N$  or  $k$  (i.e.,  $N, k \ll 10$ ) this distance is, within the numerical accuracy of the data, identical to  $R_e(N)$  with  $N = k$ . This agreement also demonstrates that the difference between a segment of a long chain and a chain having the same length as the segment becomes irrelevant for distances larger than  $\xi$ . In precisely this sense the (long-range) excluded volume interactions are screened in semidilute solutions and in melts. Mean-field descriptions become appropriate on the level of coarse-grained (Gaussian) chains of blobs.<sup>6,7</sup>

### 2.3 Dynamics of Polymer Melts: Rouse and Reptation Models

*The Rouse Model.* As a monomer in a dilute solution moves, it creates a vortex (“wave”) in the solvent. The solvent transports the “wave” which is transported to other monomers of the chains so that a coupling between the motion of (distant) monomers arises (see Ref. 14). This long-range hydrodynamic interaction becomes screened by other chains when the concentration of the solution increases.<sup>7</sup> In a dense melt, hydrodynamic interactions are completely suppressed. Thus, it is generally believed that the Rouse theory<sup>6,7</sup> provides a viable description for the long-time behavior of polymer dynamics in a melt, provided entanglements with other chains, giving rise to reptation dynamics,<sup>7,9</sup> are not important

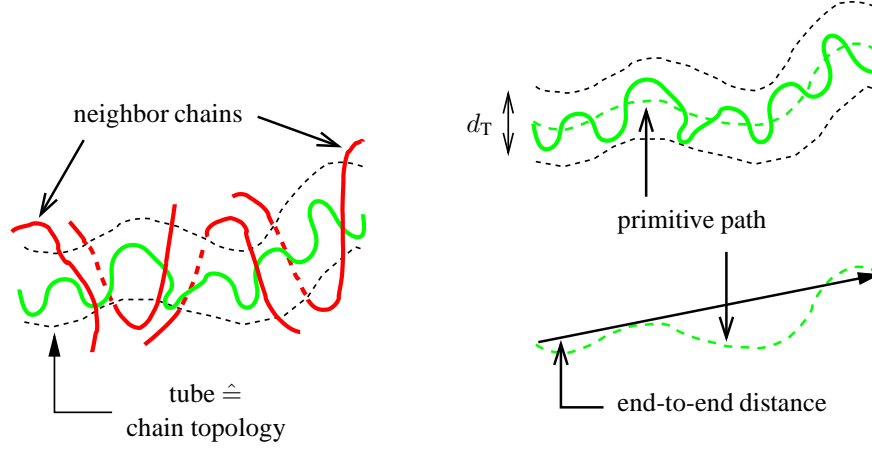


Figure 5. Sketch of the reptation concept for the dynamics of long-chain polymer melts.<sup>7,9</sup> The chain is supposed to be enclosed in a “tube” formed by its neighbors. The tube may be characterized by an axis, the primitive path. The tube confines the motion of the enclosed chain: It predominantly moves along the primitive path. Perpendicular excursions are suppressed beyond the tube diameter  $d_T$ . The tube diameter is larger than the effective bond length  $b$ :  $d_T = b\sqrt{N_e}$ , where the “entanglement length”  $N_e \gg 1$ . The primitive path represents the shortest connection between the chain ends, which respects the topology imposed on the enclosed chain by the entanglements with its neighbors. The length  $L$  of the primitive path is thus larger than  $R_e$ , which is the shortest connection between the chain ends in space.  $L$  varies linearly with  $N$ :  $Ld_T = R_e^2$  so that  $L = d_T(N/N_e)$ .

(see Fig. 5 and also below).<sup>j</sup>

The Rouse theory assumes the chains to be ideal and models them as a sequence of Brownian beads, connected by harmonic springs and subjected to a local random force and a local friction.<sup>6,7</sup> This bead-spring model is characterized by two parameters: the effective bond length  $b$  and the monomer mobility  $m$ . The mobility, or more precisely  $1/m$ , measures the time it takes a bead to diffuse over the distance  $b$ . Thus, the diffusion coefficient of a monomer is proportional to  $mb^2$ . As the center of mass (CM) of a chain does not experience any external force other than the antagonistic friction and random forces, the theory predicts that the CM diffuses freely at all times

$$g_3(t) = \left\langle [\vec{R}_{\text{cm}}(t) - \vec{R}_{\text{cm}}(0)]^2 \right\rangle = 6D_N t, \quad (12)$$

where  $\vec{R}_{\text{cm}}(t)$  denotes the position of the CM at time  $t$ . The diffusion coefficient of a chain is by a factor of  $N$  slower than that of a monomer, i.e.,

$$D_N \sim \frac{mb^2}{N}. \quad (13)$$

From Eqs. (12,13) the longest relaxation time  $\tau_N$  can be obtained. Arguing that a chain is relaxed when its CM has diffused over a distance of the order of its own size, we find

$$g_3(\tau_N) \sim D_N \tau_N \stackrel{!}{\sim} R_g^2 \sim b^2 N \quad \Rightarrow \quad \tau_N \sim \frac{N^2}{m}, \quad (14)$$

<sup>j</sup>To our knowledge, there is no established derivation of the Rouse model from a microscopic theory. For a recent attempt see Ref. 28.

where the ideality of the chain was exploited.

*Strongly Entangled Polymers and Reptation Model.* The single-chain picture proposed by the Rouse theory is supposed to be valid as long as entanglements with other chains do not dominate the polymer dynamics. This is believed to be the case for short chains, for which  $N$  is smaller than the entanglement length  $N_e$  (Fig. 5). For  $N \gg N_e$  the prevailing picture is that a chain is enclosed in a temporary “tube” formed by its neighbors. Entanglements force the enclosed chain to diffuse along the contour of the tube having a length of  $L \sim N$  (“reptation”; see Fig. 5).<sup>6,7</sup> Because the curvilinear diffusion through the tube is presumed to be Rouse-like, reptation theory predicts the relaxation time of the chain to scale with  $N$  as

$$\tau_N \sim \frac{L^2}{(mb^2/N)} \sim N^3 \quad (15)$$

so that the diffusion coefficient of the CM in space is given by [Eq. (14)]

$$g_3(\tau_N) \sim D_N \tau_N \stackrel{!}{\sim} R_g^2 \sim N \quad \Rightarrow \quad D_N \sim \frac{1}{N^2} . \quad (16)$$

Experimentally, one finds a still stronger dependence:  $\tau_N \sim N^{\approx 3.4}$  and  $D_N \sim N^{\approx -2.4}$ .<sup>k</sup>

Clearly, simulation methods which attempt to model the true physical dynamics, such as Molecular Dynamics or Monte Carlo algorithms employing local random moves, must suffer from these long relaxation times. Various alternative Monte Carlo methods have been proposed to efficiently equilibrate dense polymer melts. We will present some of these approaches (Sec. 5 and Sec. 6).

### 3 Monte Carlo Methods: A Brief Overview

In equilibrium statistical mechanics thermodynamic properties are calculated as ensemble averages over all points  $\mathbf{x}$  in a high-dimensional configuration space  $\Gamma$ .<sup>l</sup> In the canonical ensemble the average of an observable  $A(\mathbf{x})$  is given by

$$\langle A \rangle = \int d\mathbf{x} A(\mathbf{x}) P_{\text{eq}}(\mathbf{x}) = \frac{1}{Z} \int d\mathbf{x} A(\mathbf{x}) \exp[-\beta U(\mathbf{x})] . \quad (17)$$

In general, the integral cannot be solved analytically. Monte Carlo (MC) simulations provide a numerical approach to this problem by generating a random sample of configuration-space points  $\mathbf{x}_1, \dots, \mathbf{x}_m, \dots, \mathbf{x}_M$  according to some distribution  $P_s(\mathbf{x})$ .  $\langle A \rangle$  is then estimated by<sup>31–33</sup>

$$\overline{A} = \frac{\sum_{m=1}^M A(\mathbf{x}_m) e^{-\beta U(\mathbf{x}_m)} / P_s(\mathbf{x}_m)}{\sum_{m=1}^M e^{-\beta U(\mathbf{x}_m)} / P_s(\mathbf{x}_m)} = \frac{\sum_{m=1}^M A(\mathbf{x}_m) W(\mathbf{x}_m)}{\sum_{m=1}^M W(\mathbf{x}_m)} , \quad (18)$$

<sup>k</sup>This exponent varies very little –if at all– with the chemical properties of the (linear) polymer.<sup>29,30</sup>

<sup>l</sup>We assume that the momenta can be integrated out, since the observables only depend on the positions of the particles.

where we introduced the “weight”  $W(\mathbf{x}) = P_{\text{eq}}(\mathbf{x})/P_s(\mathbf{x})$ . Note that, while  $\langle A \rangle$  is a number,  $\bar{A}$  is still a random variable. Whether  $\bar{A}$  represents a good estimate for  $\langle A \rangle$  depends on the total number  $M$  of configurations used and, for a given  $M$ , on the choice of  $P_s(\mathbf{x})$ .

To see this in more detail<sup>m</sup> let us define the mean value with respect to  $P_s$  by

$$\langle (\cdot) \rangle_s = \int d\mathbf{x} (\cdot) P_s(\mathbf{x}) . \quad (19)$$

For large  $M$ , the average of  $\bar{A}$  and its variance  $\text{var}_s(\bar{A})$  may be estimated from the small-fluctuation approximations<sup>34</sup>

$$\left\langle \frac{Y}{Z} \right\rangle_s \approx \frac{\langle Y \rangle_s}{\langle Z \rangle_s} \left[ 1 - \frac{\langle YZ \rangle_s}{\langle Y \rangle_s \langle Z \rangle_s} + \frac{\langle Z^2 \rangle_s}{\langle Z \rangle_s^2} \right] , \quad (20)$$

$$\text{var}_s \left( \frac{Y}{Z} \right) \approx \frac{1}{\langle Z \rangle_s^2} \left[ \langle Y^2 \rangle_s - 2 \frac{\langle Y \rangle_s \langle YZ \rangle_s}{\langle Z \rangle_s} + \frac{\langle Y \rangle_s^2 \langle Z^2 \rangle_s}{\langle Z \rangle_s^2} \right] . \quad (21)$$

This gives<sup>n</sup>

$$\langle \bar{A} \rangle_s \approx \langle A \rangle - \frac{1}{M} [\langle WA \rangle - \langle W \rangle \langle A \rangle] , \quad \text{var}_s(\bar{A}) \approx \frac{1}{M} \langle W(A - \langle A \rangle)^2 \rangle . \quad (22)$$

Equation (22) shows that  $\bar{A}$  provides an unbiased estimate of  $\langle A \rangle$  in the limit  $M \gg 1$  unless  $\langle W \rangle \gg 1$ , i.e., unless  $P_s(\mathbf{x})$  is very different from  $P_{\text{eq}}(\mathbf{x})$ . When  $P_s(\mathbf{x})$  deviates considerably from  $P_{\text{eq}}(\mathbf{x})$ , it predominantly samples configuration-space points, which are not representative of the thermal equilibrium. One could try to compensate this inefficient sampling by making  $M$  larger and larger. However, on the one hand this renders the simulation very time-consuming. On the other hand, there is no guarantee that the maximum  $M$  one is willing (or able) to simulate suffices to outweigh the error incurred by the inadequate choice of  $P_s(\mathbf{x})$ .

Thus,  $P_s(\mathbf{x})$  should approximate  $P_{\text{eq}}(\mathbf{x})$  as closely as possible to obtain meaningful results from MC simulations. To this end, two approaches may be distinguished:<sup>34,35</sup>

1. *Static MC methods*: Static methods generate a sequence of *statistically independent* configuration-space points from the distribution  $P_s(\mathbf{x})$ . In this case one has to tune the algorithm cleverly so that the weights  $W(\mathbf{x})$  do not get out of hand. Examples how to achieve this will be discussed in Sec. 6.
2. *Dynamic MC methods*: Dynamic methods generate a sequence of *correlated* configuration-space points via some stochastic process which has  $P_{\text{eq}}(\mathbf{x})$  as its unique equilibrium distribution. In practice, this process is always taken to be a Markov process.<sup>32,33</sup> The defining property of a Markov process is that it has no “memory”. That is, the probability for the occurrence of the future configuration  $\mathbf{x}$  depends only on the present configuration  $\mathbf{x}'$  and not on the other configurations that the process visited in the past.

Dynamic MC methods have become a widely used simulation technique, to which we will also heavily refer in the following sections. So, we provide a brief introduction here (many more details may be found in Ref. 36).

<sup>m</sup>In part, our discussion closely follows Sec. 2.3 of Ref. 34.

<sup>n</sup>Note that  $\langle W \rangle_s = 1$ ,  $\langle WA \rangle_s = \langle A \rangle$ , etc. In Eq. (22) the  $\approx$ -sign means that there are corrections of  $O(1/M^2)$  which we have neglected.

Let us assume that the configuration space is discrete and that the Markov process evolves in this space in discrete time steps  $\Delta t (= 1)$ . The time evolution of this Markov chain may be characterized by the “master equation” for the probability  $P(\mathbf{x}, t)$  to find the system in the state  $\mathbf{x}$  at time  $t$

$$P(\mathbf{x}, t+1) - P(\mathbf{x}, t) = \sum_{\mathbf{x}' \neq \mathbf{x}} \left[ w(\mathbf{x}|\mathbf{x}')P(\mathbf{x}', t) - w(\mathbf{x}'|\mathbf{x})P(\mathbf{x}, t) \right]. \quad (23)$$

Here,  $w(\mathbf{x}|\mathbf{x}')$  denotes the transition probability from  $\mathbf{x}'$  to  $\mathbf{x}$  which is independent of time. (In the continuous time limit ( $\Delta t \rightarrow 0$ ) it becomes a “transition rate”, i.e., a transition probability per unit time.) Equation (23) expresses the balance between the flux of all other states  $\mathbf{x}'$  towards  $\mathbf{x}$  (first term of the rhs), leading to an increase of  $P(\mathbf{x})$ , and the flux away from  $\mathbf{x}$  (second term of the rhs) which diminishes  $P(\mathbf{x})$ . Note that only terms with  $\mathbf{x} \neq \mathbf{x}'$  contribute. We can rewrite Eq. (23) by including the missing term for  $\mathbf{x} = \mathbf{x}'$  if we take into account that  $w(\mathbf{x}|\mathbf{x}')$  is normalized. Since a transition from  $\mathbf{x}'$  to some state  $\mathbf{x}$ , including  $\mathbf{x}'$ , will occur with certainty,  $w(\mathbf{x}|\mathbf{x}')$  satisfies

$$\sum_{\mathbf{x}'} w(\mathbf{x}|\mathbf{x}') = 1. \quad (24)$$

Inserting Eq. (24) into Eq. (23) the master equation takes the following form

$$P(\mathbf{x}, t+1) = \sum_{\mathbf{x}'} w(\mathbf{x}|\mathbf{x}')P(\mathbf{x}', t). \quad (25)$$

For the application of these results to statistical physics it is necessary that  $P(\mathbf{x}, t)$  converges to a unique stationary distribution, irrespective of the initial configuration of the system, in the long-time limit and that this distribution is the (canonical) equilibrium distribution  $P_{\text{eq}}(\mathbf{x})$ . Thus, the right-hand side of Eq. (23) must vanish for  $P(\mathbf{x}', t) = P_{\text{eq}}(\mathbf{x}')$ . Certainly, this is the case if each term of the sum vanishes separately. This leads to the condition of “detailed balance” (see Refs. 31–33, 36)

$$w(\mathbf{x}|\mathbf{x}')P_{\text{eq}}(\mathbf{x}') = w(\mathbf{x}'|\mathbf{x})P_{\text{eq}}(\mathbf{x}). \quad (26)$$

To exploit this condition in MC algorithms the transition probability may be split into two independent parts: First, we propose a transition from  $\mathbf{x}'$  to  $\mathbf{x}$  according to some probability  $P_{\text{pro}}(\mathbf{x}' \rightarrow \mathbf{x})$ . Then, this move will be accepted or rejected with probabilities  $\text{acc}(\mathbf{x}' \rightarrow \mathbf{x})$  and  $1 - \text{acc}(\mathbf{x}' \rightarrow \mathbf{x})$ , respectively. So, we have

$$\frac{w(\mathbf{x}|\mathbf{x}')}{w(\mathbf{x}'|\mathbf{x})} = \frac{P_{\text{pro}}(\mathbf{x}' \rightarrow \mathbf{x}) \text{acc}(\mathbf{x}' \rightarrow \mathbf{x})}{P_{\text{pro}}(\mathbf{x} \rightarrow \mathbf{x}') \text{acc}(\mathbf{x} \rightarrow \mathbf{x}')} = e^{-\beta[U(\mathbf{x}) - U(\mathbf{x}')]}. \quad (27)$$

To solve this equation for  $\text{acc}(\mathbf{x}' \rightarrow \mathbf{x})$  we set

$$\text{acc}(\mathbf{x}' \rightarrow \mathbf{x}) = F \left( \frac{P_{\text{pro}}(\mathbf{x} \rightarrow \mathbf{x}') e^{-\beta U(\mathbf{x})}}{P_{\text{pro}}(\mathbf{x}' \rightarrow \mathbf{x}) e^{-\beta U(\mathbf{x}')}} \right). \quad (28)$$

From Eq. (27) we see that the function  $F(x)$  satisfies  $F(x)/F(1/x) = x$ . One solution to this equation was proposed by Metropolis *et al.*:<sup>37</sup>  $F(x) = \min(1, x)$ . This leads to the “Metropolis criterion” for the acceptance probability

$$\text{acc}(\mathbf{x}' \rightarrow \mathbf{x}) = \min \left( 1, \frac{P_{\text{pro}}(\mathbf{x} \rightarrow \mathbf{x}')}{P_{\text{pro}}(\mathbf{x}' \rightarrow \mathbf{x})} e^{-\beta[U(\mathbf{x}) - U(\mathbf{x}')] } \right). \quad (29)$$

The Metropolis criterion is the core of essentially all dynamic MC algorithms. It embodies detailed balance which guarantees that the simulation, irrespective of the initial configuration, converges to the canonical equilibrium distribution, provided the set of chosen Monte Carlo moves leads to ergodic sampling.<sup>o</sup>

*Detailed Balance versus Stationarity.* Detailed balance is an important, but very strict criterion. Less stringent is the condition of stationarity [Eq. (25)]

$$P_{\text{eq}}(\mathbf{x}) = \sum_{\mathbf{x}'} w(\mathbf{x}|\mathbf{x}') P_{\text{eq}}(\mathbf{x}') , \quad (30)$$

implying that  $P_{\text{eq}}(\mathbf{x})$  remains invariant under the Markov dynamics. Stationarity in conjunction with the ergodicity of chosen set of MC moves ensures a valid simulation.<sup>38,40</sup>

In practice, this milder condition may be important. Imagine that we want to update a polymer chain consisting of  $N$  monomers and that each monomer can be displaced in  $N_{\text{dis}}$  directions. One possibility is to select a monomer and a direction randomly. Thus,  $P_{\text{pro}}(\mathbf{x}' \rightarrow \mathbf{x}) = 1/(N N_{\text{dis}}) = P_{\text{pro}}(\mathbf{x} \rightarrow \mathbf{x}')$ . This procedure obeys detailed balance: In the next move the same monomer and the reverse displacement may be chosen with the same a priori probability. On the other hand, one could also attempt to move one monomer after the other, proceeding regularly from monomer 1 to monomer  $N$ . This sequential updating scheme violates detailed balance: The next step never selects again the monomer whose displacement has just been attempted. So, the probability for the reverse move is zero.

However, sequential updating is a valid scheme if the individual steps obey detailed balance<sup>40</sup> or at least stationarity. To see that we can write the transition probability from  $\mathbf{x}'$  to  $\mathbf{x}$  for sequential updating as

$$w(\mathbf{x}|\mathbf{x}') = \sum_{\mathbf{z}_N} \cdots \sum_{\mathbf{z}_2} \sum_{\mathbf{z}_1} w^{(N)}(\mathbf{x}|\mathbf{z}_N) \cdots w^{(2)}(\mathbf{z}_2|\mathbf{z}_1) w^{(1)}(\mathbf{z}_1|\mathbf{x}') . \quad (31)$$

This means that the process passes sequentially first with probability  $w^{(1)}(\mathbf{z}_1|\mathbf{x}')$  from  $\mathbf{x}'$  to  $\mathbf{z}_1$  by attempting to move the first monomer, then from  $\mathbf{z}_1$  to  $\mathbf{z}_2$  by attempting to move the second monomer, and so on until configuration  $\mathbf{x}$  is reached. Multiplying Eq. (31) by  $P_{\text{eq}}(\mathbf{x}')$  and summing over all  $\mathbf{x}'$  we find

$$\begin{aligned} & \sum_{\mathbf{x}'} w(\mathbf{x}|\mathbf{x}') P_{\text{eq}}(\mathbf{x}') \\ &= \sum_{\mathbf{z}_N} \cdots \sum_{\mathbf{z}_2} \sum_{\mathbf{z}_1} w^{(N)}(\mathbf{x}|\mathbf{z}_N) \cdots w^{(2)}(\mathbf{z}_2|\mathbf{z}_1) \underbrace{\sum_{\mathbf{x}'} w^{(1)}(\mathbf{z}_1|\mathbf{x}') P_{\text{eq}}(\mathbf{x}')}_{= P_{\text{eq}}(\mathbf{z}_1)} \\ &= \dots = P_{\text{eq}}(\mathbf{x}) , \end{aligned} \quad (32)$$

---

<sup>o</sup>By “ergodic” sampling we mean that the probability of finding the system in configuration  $\mathbf{x}$ , starting from some state  $\mathbf{x}'$  (including  $\mathbf{x}$ ), is non-zero for all  $\mathbf{x}$  after a sufficiently long time.<sup>38</sup> This definition is a bit dangerous because it conflicts with others in the literature. For instance, in mathematical texts on Markov chains (= discrete-time Markov processes with a discrete configuration space) our definition rather corresponds to an “irreducible and aperiodic” chain (there, “ergodic” is a synonym for “irreducible”).<sup>34,35,39</sup> In Ref. 40 our definition would be termed “regular sampling”.

i.e., sequential sampling preserves the stationarity of the equilibrium distribution. Thus, it is a correct simulation procedure. This conclusion is important for a variety of MC methods which perform different trial moves in a fixed order.

## 4 Some Coarse-Grained Simulation Models

In Sec. 1 we introduced the term “coarse-grained model”. This was defined as a model which associates a group of chemical monomers with a “bead” (effective monomer) in order to eliminate microscopic degrees of freedom (bond length vibrations, etc.). Here, we refine our definition and distinguish between two types of coarse-grained models:

1. *The coarse-grained model is derived from a specific polymer.* In practice, this usually implies that the properties of the model (potential parameters, density, etc.) have to be adjusted to results from atomistic simulations of the polymer under consideration (see Appendix A for an example). The incentive to devise such models rests upon the fact that they may be simulated much more efficiently than their atomistic counterpart. Thus, it is tempting to split the simulation into two levels: First, one uses the coarse-grained model for equilibration and for the determination of large-scale properties. Then, atomistic details may be reinserted to allow for a thorough comparison with experiments. Recent attempts to perform such multi-scale approaches are described in Refs. 41, 42 (see also Appendix A).
2. *The coarse-grained model has no direct connection to any specific polymer.* It is a generic model retaining only features common to all polymers of the same chain topology. For (uncharged) linear polymers these features are chain connectivity, excluded-volume interactions, and, additionally, monomer-monomer attractions if one wants to simulate  $\Theta$ - or bad-solvent conditions (see Fig. 3). Many of these generic models, be it lattice or continuum models, have been introduced in the literature (see Refs. 43, 44 for comprehensive overviews). In the following we present those models in more detail, which will be discussed in Secs. 5, 6.

### 4.1 Lattice Models

*The Self-Avoiding Walk.* About 50 years ago Orr and Montroll<sup>46</sup> proposed the self-avoiding walk (SAW) as a model for a linear polymer in a good solvent. The SAW is defined on a discrete lattice, often on a square or simple cubic lattice (Fig. 6). Each monomer occupies one lattice site, the bond length equals the lattice constant, and the bond angles are restricted by the lattice geometry and by the repulsive hard-core monomer-monomer interaction (e.g.  $90^\circ$  and  $180^\circ$  for the cubic lattice, as immediate backfolding is forbidden). This model can be complemented by attractive interactions if, for instance, an energy gain  $-\epsilon$  is associated with every occupied nearest neighbor pair.<sup>47</sup> In addition to excluded volume interactions the simulation then also has to take account of the Boltzmann factor  $\exp(n_{\text{nn}}\epsilon/k_{\text{B}}T)$ , where  $n_{\text{nn}}$  is the number of nearest neighbors.

To simulate the SAW by dynamic Monte Carlo one must first decide about the elementary moves that propose a new SAW configuration  $\mathbf{x}$  from an old one  $\mathbf{x}'$ . The earliest suggestion<sup>48</sup> comprised one-bead excitations<sup>34, 43, 44</sup> (Fig. 6). In these algorithms, one

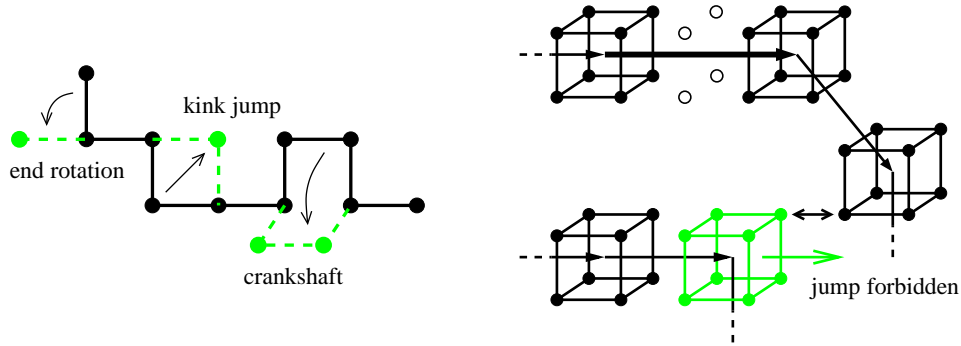


Figure 6. *Left figure*: Single-site self-avoiding walk (SAW) of chain length  $N = 10$  on a simple cubic lattice (solid lines and black dots). The grey dots and the grey dashed lines indicate the moves discussed in the text: end-bond rotation, kink jumps and  $90^\circ$  crankshaft motion. *Right figure*: Sketch of a possible configuration of monomers in the 3D bond-fluctuation model (BFM). (A vectorized version of the BFM algorithm can be found in Ref. 45.) The bond vector  $(3, 0, 0)$  (thick black arrow) blocks four lattice sites (marked by  $\circ$ ) that are no longer available to other monomers due to the excluded volume interaction. This interaction also prevents the jump of the grey monomer in the direction of the large arrow ( $\longrightarrow$ ), since the corners of the monomers, indicated by  $\leftrightarrow$ , would then occupy the same lattice site.

chooses a monomer at random. If the monomer is at the chain end, the bond to its neighbor is turned to a randomly selected lattice direction. Due to the fixed bond length an inner monomer is only mobile if its bond angle is  $90^\circ$  on the square or simple cubic lattice. In this case, one attempts a “kink-jump” motion, i.e., a one-bead flip to the opposite lattice site. End-bond rotation and kink jumps are accepted according to the Metropolis criterion if the target sites are empty.

These moves are special examples of the class of “local  $N$ -conserving moves”.<sup>34</sup> Quite generally, a “local move” alters the configuration of a small piece of the original SAW while leaving the remaining monomers unchanged. This definition opens the possibility to invent moves comprising more than one bead, such as two-bead or three-bead excitations. Figure 6 shows a common example, the  $90^\circ$  crankshaft motion (only possible in 3D). The crankshaft motion removes an important drawback of kink jumps. It introduces new bond vectors, whereas a kink jump does not. Therefore, if only end-bond rotations and kink jumps are allowed, new bond orientations have to diffuse from the ends toward the interior part of the chains. This algorithm is not very efficient in reshuffling the bond vectors and so in preparing independent configurations. The inclusion of crankshaft motions remedies this problem.

However, even then a disturbing feature remains. It has been proved that all local  $N$ -conserving algorithms for two- and three-dimensional SAW’s are not ergodic for large  $N$ .<sup>49</sup> There are dense configurations (“double cul-de-sac” in 2D, “knots” in 3D; see Ref. 34) which are completely frozen: They can neither be transformed into nor reached from other configurations. Whether this problem is serious in practice is a question that, to our knowledge, is not fully settled (see e.g. Ref. 43 or footnote 9 of Ref. 50). One can argue that, if one starts from an extended configuration—for instance, from a straight rod—and if one is interested in high- $T$  properties only, non-ergodicity effects due to compact structures should be small. This argument may be true for short chains,<sup>p</sup> but should fail for long

<sup>p</sup>Here, it is not clear what “short” really means. For  $N \lesssim 10^2$  the error incurred by using local  $N$ -conserving



ones, since it has been proved that the fraction of SAW's belonging to the ergodicity class<sup>q</sup> of the straight rod is exponentially small in the large- $N$  limit. Of course, if one is interested in low- $T$  properties, problems with non-ergodicity might be sizable, even for small chain length.<sup>51</sup>

*The Bond-Fluctuation Model.* The bond-fluctuation model (BFM) was proposed<sup>52,53</sup> as an alternative to a (single-site) SAW model, which retains the computational efficiency of the lattice without being plagued by severe ergodicity problems. The key idea is to increase the size of a monomer which now occupies, instead of a single site, a whole unit cell of the lattice (e.g. a square for the 2D- or a cube for the 3D hyper-cubic lattice; see Fig. 6). This enlarged monomer size has two important consequences:

1. A priori, many different bond vectors can occur. This multitude is restricted by two conditions. First, adjacent monomers may not overlap. This limits the bond length to  $\ell \geq \ell_{\min} = 2$  (in units of the lattice constant). Second, the hard-core monomer-monomer interaction should suffice to prevent two bonds from intersecting each other in the course of the simulation. In 2D this only imposes an upper bound on the bond length,  $\ell \leq \ell_{\max} = \sqrt{13}$ ,<sup>52,53</sup> whereas in 3D, in addition to  $\ell \leq \ell_{\max} = \sqrt{10}$ , some smaller bond vectors also have to be excluded.<sup>54</sup> The resulting sets of allowed bond vectors are:

$$\begin{aligned} \{\vec{b}\} &= [2, 0], [2, 1], [2, 2], [3, 0], [3, 1], [3, 2] \quad (2D) , \\ \{\vec{b}\} &= [2, 0, 0], [2, 1, 0], [2, 1, 1], [2, 2, 1], [3, 0, 0], [3, 1, 0] \quad (3D) , \end{aligned} \quad (33)$$

where  $[\cdot]$  denotes a class of bond vectors sharing the same length, but differing in direction. For instance, the class  $[2, 0]$  ( $[2, 0, 0]$ ) comprises all vectors with a length of 2 and direction along the lattice axis (4 directions in 2D, 6 in 3D). Equation (33) gives rise to 41 bond angles in 2D<sup>55</sup> and to 87 bond angles in 3D.<sup>54</sup> This has to be compared to 3 (2D) or 5 (3D) bond angles for the SAW model on the hypercubic lattice where a monomer is associated with a lattice site. Due to the multitude of different bond lengths and bond angles the BFM is much closer to continuous-space behavior than the single-site lattice model<sup>r,57</sup>.

2. Ergodicity problems are much less severe than for the single-site SAW. For the BFM a local  $N$ -conserving move consists of selecting a monomer at random and of attempting a displacement by one lattice constant in a randomly chosen lattice direction. As these local jumps<sup>s</sup> permit transitions between different vectors, the algorithm can escape from configurations where a single-site model would be frozen in.<sup>52</sup> If the attempted displacement satisfies both the bond vectors constraints of Eq. (33) and the excluded volume interaction, the move is accepted. Of course, it is also possible

---

algorithms seems to be small (see Ref. 43 and the footnote 9 of Ref. 50).

<sup>q</sup>By "ergodicity class of a straight rod" we mean all mutually accessible configurations, one of which is the rod.  
<sup>r</sup>The main advantage of lattice models is their computational efficiency. Longer length and times scales may be probed. However later on, the results of the simulation shall be compared to theories or experiences, which "live" in continuous space. So, the important question arises of how well the lattice algorithm approximates continuum properties. A general, intuitive answer is: The finer the lattice, i.e., the more sites are occupied by one particle, the closer the continuum limit should be realized. Recently, this statement was made more precise by the example of monatomic fluids interacting via a Lennard-Jones or a Buckingham potential.<sup>56</sup>

<sup>s</sup>Larger jumps distances were also tested (in 3D), but found less efficient in concentrated solutions.<sup>54</sup>

to include a finite interaction energy. Then, the move is accepted according to the Metropolis criterion. A possible choice<sup>t</sup> is to work with an energy  $-\epsilon$  between pairs of monomers with distance  $2 \leq r \leq \sqrt{6}$ . This interval comprises all neighbors which contribute to the first peak of the pair-distribution function<sup>59</sup> in a dense polymer system.<sup>54</sup> This choice was made in studies of the  $\Theta$ -point<sup>60</sup> and of the phase transition in binary polymer blends (see Ref. 11).

## 4.2 Continuum Models

*Two Bead-Spring Models.* A widely used continuum model is the bead-spring model introduced by Grest and Kremer.<sup>61</sup> In this model nearest-neighbor monomers along the backbone of the chain are bonded to each other by a FENE (finitely extendible non-linear elastic) potential

$$U_F(\ell) = \begin{cases} -\frac{1}{2}k\ell_{\max}^2 \ln [1 - (\ell/\ell_{\max})^2] & \ell \leq \ell_{\max} , \\ \infty & \text{else} , \end{cases} \quad (34)$$

whereas all monomers, bonded and non-bonded ones, interact via a truncated and shifted Lennard-Jones (LJ) potential

$$U_{\text{LJ}}^{\text{ts}}(r) = \begin{cases} 4\epsilon [(\sigma/r)^{12} - (\sigma/r)^6] + C(r_{\text{cut}}) & \text{for } r \leq r_{\text{cut}} , \\ 0 & \text{else} , \end{cases} \quad (35)$$

where  $C(r_{\text{cut}})$  ensures that the potential vanishes at the cut-off parameter  $r_{\text{cut}}$ . Such a cut-off is commonly employed to render the interaction short-ranged (Fig. 7).<sup>u</sup> The parameter  $\epsilon$  defines the energy scale and  $\sigma$  the length scale of the system. That is, we set  $\epsilon = \sigma = 1$  (LJ units) in the following.

For small values of the bond length the FENE potential is harmonic (“elastic behavior”), i.e.,  $U_F(\ell) = k\ell^2/2$  for  $0 \leq \ell \ll \ell_{\max}$ , whereas the logarithmic divergence imposes  $\ell < \ell_{\max}$  (“finite extensibility”). The parameters  $\ell_{\max}$  and  $k$  have to be chosen such that the possibility of bond crossing becomes so unlikely that it never occurs. Reference 61 suggests  $k = 30$  and  $\ell_{\max} = 1.5$  (in LJ units). This has become a standard choice.

The FENE potential alone does not prevent monomers from overlapping. To realize excluded volume the LJ-interaction has to be taken into account also between bonded monomers. The superposition of the FENE- and the LJ-potentials yields a steep effective bond potential with a minimum at  $\ell_0 \simeq 0.96$  (Fig. 7). The shape of the bond potential depends on the cut-off parameter of the LJ-interaction:

- If one takes  $r_{\text{cut}} = r_{\min} = 2^{1/6}$  ( $C(r_{\text{cut}}) = 1$ ), i.e., as the minimum of the LJ-potential, the monomer-monomer interaction becomes purely repulsive. This model is commonly called “Kremer-Grest model”.<sup>61</sup> For isolated chains it realizes good solvent conditions.

<sup>t</sup>Another choice uses a discretization of the Lennard-Jones potential.<sup>58</sup>

<sup>u</sup>From a computational point of view short-range interactions are convenient because the simulation can be speeded up by neighbor lists.<sup>31,62</sup> However, as the truncation ignores the contribution of the tail of the potential, the error incurred must be corrected before comparing with results for the full potential. For instance, the truncation shifts the location of the critical point of the liquid-gas transition in a LJ-liquid (see Ref. 63 or Sec. 3.2.2 of Ref. 31 for details). To avoid these truncation effects some authors prefer to work with the full LJ-potential.<sup>64</sup>

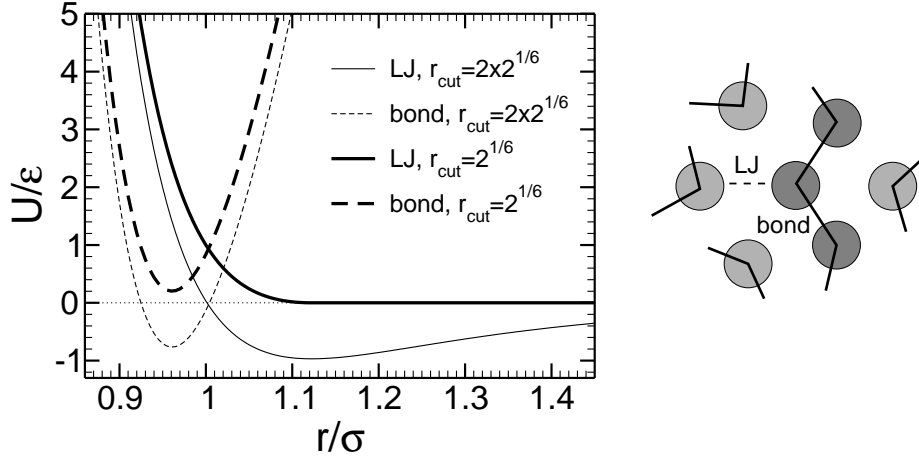


Figure 7. Bond and Lennard-Jones potentials versus the distance  $r$  between two monomers of the bead-spring model (for the bond potential  $\vec{r} = \vec{b}$ ). The bond potential results from the superposition of Eqs. (34,35). For both cut-off parameters  $r_{\text{cut}}$  the bond potential was shifted by  $-20$  to show it on the same scale as the LJ-potentials. The LJ-potential with  $r_{\text{cut}} = 2^{1/6}$  is purely repulsive, whereas the potential with  $r_{\text{cut}} = 2 \times 2^{1/6}$  has an attractive minimum at  $r_{\text{min}} = 2^{1/6}$ .

- The simulation of  $\Theta$ - or bad solvents requires to incorporate part of the LJ-attraction by increasing  $r_{\text{cut}}$ . Obviously, there is freedom where to cut off the attractive part. One possibility is  $r_{\text{cut}} = 2 \times 2^{1/6}$  ( $C(r_{\text{cut}}) = 127/4096$ ).<sup>65</sup> This choice is a compromise between the wish to include the major part of the attractive interaction and the need to keep the potential short-ranged. The resulting phase diagram was studied in Ref. 66.

*(Yet) Another Bead-Spring Model.* If we recall the idea of the coarse-graining –a coarse-grained monomer stands for a group of chemical monomers– it appears plausible that coarse-grained monomers are softer than their chemical counterparts. Thus, an exponent smaller than 12 in Eq. (35) may be better suited to represent their repulsion. In fact, such an observation was made in a recent effort to develop a coarse-grained model for poly(vinyl alcohol) (see Ref. 67 and Appendix A). This study also suggests the following generic model which may be considered as a variant of the Kremer-Grest model.

In this (“Kremer-Grest-like”) model non-bonded monomers interact via a purely repulsive 9-6 LJ-potential,

$$U_{9-6}^{\text{rep}}(r) = \begin{cases} \epsilon_0 [(\sigma_0/r)^9 - (\sigma_0/r)^6] + C(r_{\text{min}}) & \text{for } r \leq r_{\text{min}} = (3/2)^{1/3} \sigma_0, \\ 0 & \text{else,} \end{cases} \quad (36)$$

where  $\epsilon_0 = 1.511$  and  $C(r_{\text{min}}) = 4\epsilon_0/27$ . These non-bonded interactions are excluded between nearest neighbors in the chain, which are connected to each other by a harmonic potential

$$U_{\text{bond}}(\ell) = \frac{1}{2}k(\ell - \ell_0)^2 \quad (k = 2141.84 \sigma_0^{-2}, \ell_0 = 0.97 \sigma_0). \quad (37)$$

The equilibrium bond length  $\ell_0$  agrees with that of the Kremer-Grest model. The spring constant  $k$  has to be chosen so large to inhibit bond crossings (see Ref. 68 for further

discussion). A similar bond potential, in conjunction with Eq. (35) and  $r_{\text{cut}} = 2^{1/6}$ , has recently been used to study the effect of the bond length on the structure and dynamics of polymer melts.<sup>69</sup>

*Local Moves for Continuum Models.* The continuum models are constructed for use in Molecular Dynamics simulations. However, simulation within Monte Carlo schemes is also possible. Similarly to the lattice models a local updating scheme can be realized by selecting a monomer and a direction at random and by attempting a displacement in the chosen direction. This proposition is again accepted according to the Metropolis criterion.

The size  $\Delta$  of the displacement is a tunable parameter. It should neither be too small nor too large. If  $\Delta$  is too small, many moves may be accepted, but the system advances only slowly in configuration space. Many displacements are thus needed to obtain well decorrelated configurations. On the other hand, if  $\Delta$  is too large, many moves will be rejected and the decorrelation is also slow. A scheme how to optimize the choice for  $\Delta$  is explained in Sec. 3.3 of Ref. 31.

## 5 Monte Carlo Methods for Polymers: From Local to Non-Local Moves

The method of importance sampling is based on a Markov process in configuration space. A priori, this stochastic dynamics is merely a numerical algorithm, aiming at an efficient sampling according to  $P_{\text{eq}}(\mathbf{x})$ . It need not correspond to the physical dynamics of the (polymer) system under consideration. An appealing consequence of this feature is the freedom to invent clever MC moves which decorrelate the configurations in the smallest (CPU) time possible. These non-physical moves serve to rapidly equilibrate the system and to produce statistically independent equilibrium configurations for the study of structural and thermodynamic properties. We will pursue this idea in Secs. 5.2, 5.3. In the following section we rather want to concentrate on local moves and the ensuing dynamic interpretation of the MC method.

### 5.1 Local Moves: Studying Dynamic Properties with Monte Carlo

By employing non-local moves we can explore the statics of the system, but information about its dynamic properties is lost. Of course, the equilibrated configurations could be used in a Molecular Dynamics (MD) simulation to analyze the dynamic properties. However, if one is not willing to do that, the question arises of under which conditions the MC dynamics can be realistic. The answer to this comprises two parts:

1. Certainly, one can only expect the MC dynamics to become reliable on length and time scales where the deterministic motion of the monomers has been damped by the interaction with the surrounding (other monomers and/or solvent). For instance, in a (classical) MD simulation the monomers move ballistically at early times, i.e., their displacement is proportional to  $t$ . This is a consequence of the underlying Newtonian dynamics in the limit of vanishing force. At short times the monomers behave as if they did not “feel” the bonding to their neighbors and the presence of other particles, that is, as if they were free particles. As time increases, the interaction with

the surrounding becomes important. The motion of the monomers is then a result of a multitude of individual collisions. This “averaging” over fast degrees of freedom gradually lends a stochastic character to the dynamics which ultimately becomes diffusive in the long-time limit.

2. The moves should be “physical”. Usually, this implies that they are local.<sup>v</sup> Furthermore, the dynamics should not be dominated by the momenta which are absent in Monte Carlo. The latter condition is satisfied in dense melts, but not in dilute solution. In dilute solution the motion of distant monomers along the chain backbone are coupled via hydrodynamic interactions (see Ref. 14). and Sec. 2.3). Thus, we might expect that a local Monte Carlo algorithm reproduces Rouse dynamics where these long-range interactions are neglected.

This expectation can be verified by estimating the scaling of local MC algorithms with  $N$ . To this end, let us assume that the center of mass (CM) of an isolated chain, be it on a lattice or in the continuum, may be considered as a free Brownian particle. This is reasonable, since the CM does not experience any external force other than the random force of the heat bath (resulting from the compound effect of the random monomer hops and the acceptance criterion). So, it should diffuse freely [Eq. (12)]. The corresponding diffusion constant  $D_N$  depends on chain length. To estimate this dependence we can argue that the center of mass is displaced by  $\sim b/N$ , if one monomer moves over a distance of order  $b$  while the other monomers remain fixed. This elementary motion takes on average the time  $1/m$  with  $m$  denoting again the (temperature, density, etc. dependent) mobility of the monomer. For the CM to diffuse over the distance  $b$ ,  $N$  such random motions are needed. This takes the time  $m \times N$ , which we use as our time unit here.<sup>w</sup> Utilizing Eq. (12) we then find  $g_3(t=1) \sim (mN) \times (b/N)^2 \sim D_N$ . So,

$$D_N \sim \frac{mb^2}{N^{\approx 1}}. \quad (38)$$

Inserting this result in Eq. (14) we obtain the relaxation time of a chain

$$\tau_N \sim \frac{N^{\approx 1+2\nu}}{m} \sim \begin{cases} N^{\approx 2} & (\text{ideal chain: } \nu = 0.5) , \\ N^{\approx 2.176} & (3\text{D excluded-volume chain: } \nu = 0.588) . \end{cases} \quad (39)$$

Equations (38,39) agree with the predictions of the Rouse theory [Eqs. (13,14)].

*Monte Carlo Dynamics versus Molecular Dynamics: An Example.* The previous arguments suggest that the MC dynamics, based on local moves, becomes realistic for time and length scales outside the microscopic regime (of a bond). We want to support this assertion by a comparison between MC and MD simulations.

Figure 8 shows the diffusion coefficient  $D_N$  of a chain versus chain length.  $D_N$  was derived from the long-time limit of Eq. (12) for both the BFM, simulated via MC, and the Kremer-Grest-like continuum model of Eqs. (36,37), simulated via MD. The figure

<sup>v</sup>Examples for local moves of lattice models are given in Fig. 6. See the very end of Sec. 4.2 for a brief discussion of local moves in continuum models.

<sup>w</sup>This statement introduces the time unit  $\tau_{\text{mcs}}$  of a “Monte Carlo step (MCS)”. A MCS is defined as the time it takes to give each of the  $N$  monomers the possibility to move once.<sup>32,33</sup> Thus, we measure time in units of attempted elementary moves per monomer.

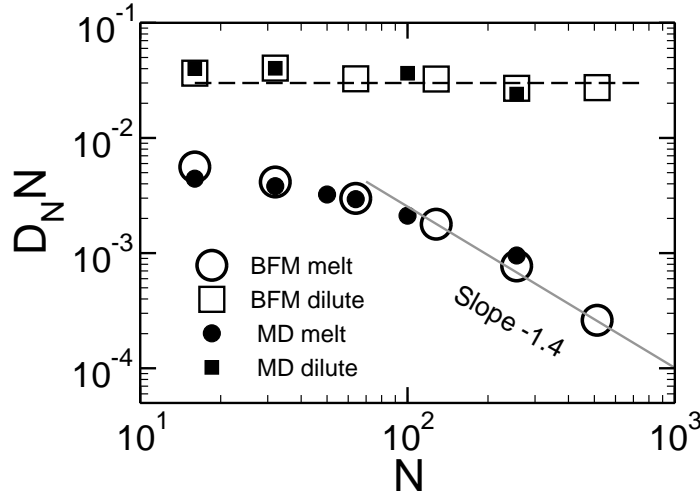


Figure 8. Diffusion coefficient  $D_N$  versus  $N$ . Two simulation methods are compared: The open symbols represent MC data of the (lattice) BFM, the filled symbols were obtained from MD simulations of a (continuum) Kremer-Grest-like model [Eqs. (36,37)]. For both models results obtained in 3D for a dilute solution and for a melt are shown. For the BFM this corresponds to the following volume fraction  $\phi$ :  $\phi = 0.0078$  (dilute),  $\phi = 0.5$  (melt). For the Kremer-Grest-like model this corresponds to the following monomer densities  $\rho$ :  $\rho = 0.0835$  (dilute),  $\rho = 0.835$  (melt). Qualitatively, the MD simulations yield the same dependence of  $D_N$  on  $N$ . To illustrate this agreement the MD data were vertically shifted by an amount that optimizes the agreement with the MC results. (The shift factors are different for the dilute solution and the melt.) In dilute solution, we find Rouse behavior [Eq. (13)] for both methods. In the melt, the chains diffuse more slowly. The dependence of  $D_N$  on  $N$  is qualitatively compatible with the Rouse-to-reptation crossover when  $N$  passes the threshold  $N_e$  (Sec. 2.3). Quantitatively however, there are deviations. Particularly for large  $N$ , the decrease of  $D_N$  appears to be stronger than predicted by reptation theory [Eq. (16)]. Roughly, we find  $D_N \sim N^{-2.4}$ . Adapted from Ref. 70.

displays the results of the simulations for a dilute solution and a dense melt.<sup>x</sup> Clearly, there is a high degree of accord between the results, illustrating that the BFM with local moves reproduces the realistic dynamics of the MD simulations. Thus, MC simulations can be more than just a versatile tool to calculate high-dimensional integrals. They may provide information on the dynamics of a system.<sup>y</sup>

<sup>x</sup>In the BFM, density is commonly specified in terms of the volume fraction  $\phi$  of lattice sites occupied by monomers. As a monomer comprises all sites of a unit cube, the monomer density  $\rho$  is smaller than  $\phi$  by a factor 8,  $\rho = \phi/8$ . Although the value  $\phi = 0.5$  appears small, the work by Paul *et al.*<sup>27</sup> established that the chains have melt-like properties at this density (see also Ref. 4). Since then,  $\phi = 0.5$  has become a standard choice (in 3D). For the Kremer-Grest model, the work of Ref. 61 showed that a monomer density of  $\rho = 0.85$ , or a value close to this, is a good choice to realize melt-like behavior. We adopted this choice in our MD study. The MD simulations were done at constant volume and constant temperature (Langevin thermostat<sup>62</sup>).

<sup>y</sup>There is further ample evidence for the correctness of this statement from other studies. For simple liquids of LJ-particles see e.g. Ref. 71. For polymers see the review in Ref. 8 or the comparison of MD simulations for polybutadiene and polyethylene with MC simulations of the BFM.<sup>72</sup> Furthermore, Monte Carlo methods have been applied to simulate dynamic processes in such diverse fields as relaxation phenomena in spin and structural glasses, spinodal decomposition of mixtures, nucleation processes, diffusion-limited aggregation, etc. (see e.g. the textbooks of Binder and Heermann<sup>32</sup> or Landau and Binder<sup>33</sup>).

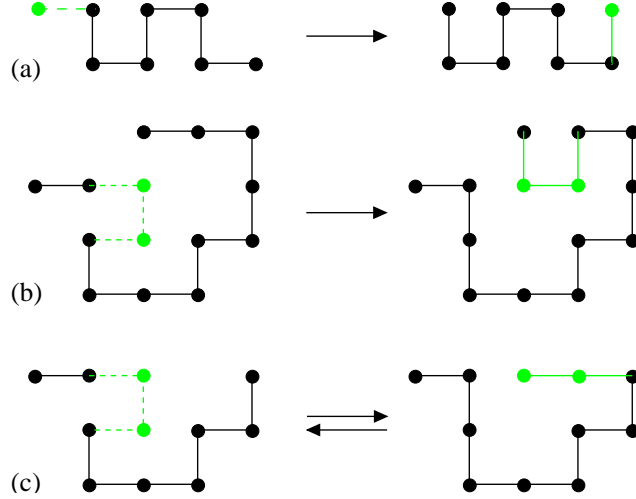


Figure 9. Slithering-snake move (a) and general reptation moves (b,c). Both moves are illustrated by the shrinkage-growth implementation. For the slithering-snake algorithm, a randomly chosen end bond (dashed line) is removed and then a new bond vector (also randomly chosen) is attached to the other chain end. For the general reptation algorithm, three moves are shown: Kink-kink transport (b) and kind-end/end-kink reptation (c). Kink-kink transport implies that a randomly chosen kink is shrunk to a bond and a new kink is inserted somewhere else along the chain. Kink-end reptation ( $\rightarrow$ ) amounts to replacing a randomly chosen kink by a bond and to appending two new bond vectors (also randomly chosen) to the other chain end. End-kink reptation ( $\leftarrow$ ) corresponds to the reverse “reaction”.

*Relaxation Time and Computational Complexity.* An important issue in any algorithm is its “computational complexity”. Quite generally, the computational complexity may be defined as the time required to solve a computational problem.<sup>73</sup> Here, the computational problem is to decorrelate chain configurations. According to Eq. (39) this takes a relaxation time  $\tau_N \sim N^{1+2\nu}$  in units of the Monte Carlo step (MCS; see footnote on page 21). As a MCS comprises  $N$  attempted moves of a monomer, the computational complexity  $\tau_{cc}$  scales with  $N$  as  $\tau_{cc} = N\tau_N \sim N^{2+2\nu}$ .

This rapid increase of  $\tau_{cc}$  with chain length—called “critical slowing-down”<sup>33</sup>—makes it difficult in practice to efficiently decorrelate configurations of long chains by local moves. In order to be able to simulate large chains with sufficient statistics, moves have to be implemented, which reduces ( $\tau_{cc} \sim N^\alpha$  with  $\alpha < 2 + 2\nu$ ) or even eliminate ( $\tau_{cc} \sim N^0$ ) the critical slowing-down. These moves cannot be local, they have to act, in some way, on all monomers of the chain. In the following we want to discuss two examples of such global updates: bilocal moves and the pivot algorithm.

## 5.2 Bilocal Moves: The Slithering-Snake and the Extended Reptation Algorithms

A bilocal  $N$ -conserving move consists in altering the configuration of two small groups of consecutive monomers. The groups are usually far from one another along the backbone of the chain. Typical examples are the slithering-snake and the extended reptation algorithm:

- The *slithering-snake* (or *reptation*) algorithm removes a bond from one chain end,

adds a new one to the other end and shifts the inner monomers one bond up along the chain in direction of the new bond (Fig. 9a). As the positions of the inner monomers remain unchanged, the chains “slither” along its contour during the MC move (whence the name of the algorithm).<sup>z</sup>

- The *extended reptation* algorithm transports a kink or an end group via a slithering motion along the chain.<sup>a</sup> Commonly utilized moves are: (1) “Kink-kink reptation”, which deletes a kink at some position along the chain and inserts a new one at another position (Fig. 9b). (2) “Kink-end reptation”, which removes a kink somewhere along the chain and adds two new bonds at one of the chain ends (Fig. 9c→). (3) “End-kink reptation”, the inverse of “kink-end reptation” (Fig. 9c←).

In the remainder of this section we will concentrate on the slithering-snake algorithm. Extended reptation is only discussed in comparison to the slithering-snake algorithm.

*Implementation and Ergodicity.* The slithering-snake and the extended reptation algorithms can be implemented in two ways: in a shrinkage-growth or a growth-shrinkage fashion. As growth-shrinkage is just the inverse of shrinkage-growth, we illustrate the procedure for the latter via the example of an isolated chain.<sup>b</sup> For the slithering-snake algorithm one chain end is selected at random, the bond to its neighbor is cleaved, and a randomly chosen new bond vector is attached at the other end. If this move respects the excluded-volume condition in the athermal case and additionally passes the Metropolis test in the thermal case, it is accepted. Otherwise the old configuration is recounted. For the extended reptation algorithm the procedure is more complicated. Details may be found for the SAW on a hypercubic lattice in Refs. 51, 76 and for a continuum bond-fluctuation model in Ref. 77.

Usually, shrinkage-growth is preferred to the growth-shrinkage procedure because it is computationally more efficient. The reason for this is illustrated in Fig. 10. The nested configuration of Fig. 10a would be frozen, if a new bond had to be appended before an end bond may be removed. However, it can be unraveled when shrinkage is attempted first. Thus, the shrinkage-growth algorithm is less plagued by –though not exempt of– non-ergodicity effects. An example is provided by the double cul-de-sac configuration of Fig. 10b.<sup>c</sup> It is frozen in the shrinkage-growth procedure, but not for the kink-end reptation move shown in Fig. 10c. In fact, kink-end/end-kink moves are known to be ergodic<sup>76</sup> (as well as other bilocal algorithms; see Ref. 50 for a thorough discussion).

Should one thus abandon the slithering-snake algorithm in favor of extended reptation? Usually, the answer is “No”. For the SAW on the hypercubic lattice problems with ergodicity arise to the constraints imposed by the small coordination number of the lattice. If

<sup>z</sup>The slithering-snake algorithm was invented by Kron in the 1960’s and later independently by Wall and Mandel.<sup>74</sup> For an overview of applications to SAW’s see e.g. Ref. 43 and to off-lattice models see e.g. Ref. 75.

<sup>a</sup>This generalization of the slithering-snake algorithm was first discussed in detail by Reiter.<sup>76</sup> More recently, algorithmic and statistical properties of extended reptation moves were analyzed and their implementation was discussed in Refs. 50, 51.

<sup>b</sup>For the multi-chain system the only difference to the isolated chain is that additionally one chain out of the  $n$  chains in the systems has to be chosen at random.

<sup>c</sup>Non-ergodicity effects are less severe for the slithering-snake algorithm than for the  $N$ -conserving local moves discussed before. For the slithering-snake algorithm the ergodicity class of a straight rod contains at least a fraction of  $N^{-(\gamma-1)/2}$  of all SAW configurations, whereas this fraction is exponentially small for the local algorithms.<sup>34,51</sup>



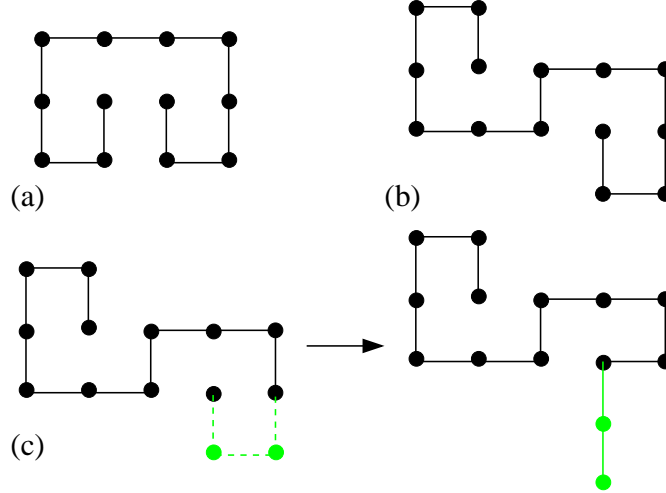


Figure 10. Configurations of 2D SAW's to illustrate the ergodicity problem of the slithering-snake algorithm (a,b) and its solution via extended reptation moves (c). Panel (a) shows a configuration that cannot be moved by slithering-snake moves, if chain growth is attempted first. However, it is not blocked in the shrinkage-growth scheme. By contrast, the configuration of panel (b) is frozen for both growth-shrinkage and shrinkage-growth moves. Panel (c) shows that this configuration may be dissolved by extended reptation moves, e.g. by kink-end reptation if the chain end, where the kink is, happens to be selected for the attachment of the two bonds.

many more bond vectors are a priori possible, as for the bond-fluctuation model or for (typical) continuum models, non-ergodicity should not represent a problem.<sup>d</sup>

*Relaxation Time and Computational Complexity: Isolated Chains.* One expects that the slithering-snake algorithm is able to decorrelate configurations more efficiently than a local updating scheme, the speed-up factor being roughly of order  $N$ . This expectation results from the following heuristic argument: The elementary move of the algorithm may be interpreted as a shift of all monomers along the contour of the chain. For the CM this curvilinear motion has two consequences: (1) The curvilinear diffusion coefficient  $D_c$  should not depend on  $N$ , since all monomers are always shifted at once, irrespective of chain length. Thereby, the slithering-snake algorithm gains a factor of  $N$  in regard to the physical reptation dynamics, in which the curvilinear displacement is Rouse-like (Sec. 2.3). (2) An elementary move displaces the CM by  $\sim b$  along the chain backbone. After  $N$  such moves, the CM has diffused curvilinearly a distance of the order of the contour length  $L \propto Nb$ . Thus, the relaxation time  $\tau_N$  should be given by

$$\tau_N = \frac{L^2}{D_c} \Rightarrow \tau_N \sim N^{\approx 2}. \quad (40)$$

<sup>d</sup>See pp. 283/284 of Ref. 43 for further discussion of that point. Contrary to SAW's, the equilibrium configurations of collapsed chains are typically (very) dense. Quite generally with increasing density, the slithering-snake or the extended reptation algorithm become less efficient, as the "free volume" to add new bonds decreases (see e.g. Ref. 78 for a comparison of various algorithms to simulate high-density polymer systems and the subsequent discussion). However, the recent study of Ref. 51 for 2D SAW's with  $N \leq 3200$  at the  $\Theta$ -point suggests that extended reptation is almost as efficient as for pure SAW's with no attractive interactions.

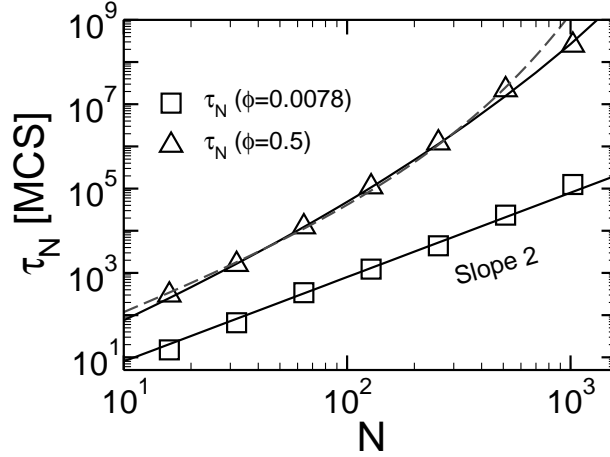


Figure 11. Relaxation time  $\tau_N$  versus  $N$ .  $\tau_N$  is defined by  $g_3(\tau_N) = R_g^2$ . In the dilute limit ( $\phi = 0.0078$ )  $\tau_N \sim N^2$ , as expected from Eq. (40). In the melt ( $\phi = 0.5$ ), the increase of  $\tau_N$  with  $N$  is stronger. The stretched exponentials are motivated by the activated reptation hypothesis:<sup>79,82,83</sup>  $\tau_N \approx N^2 \exp(0.8N^{1/3})$  (bold line) provides a better description than  $\tau_N \approx N^2 \exp(0.074N^{2/3})$  (dashed grey line). Adapted from Ref. 80.

With respect to the computational complexity (page 22) one expects  $\tau_{cc} \propto \tau_N$ . There is no extra factor of  $N$ , as in the case of local moves, for the slithering-snake algorithm. The algorithm is bilocal. It takes a time of order 1 to check and update the chain ends.<sup>e</sup>

Note that Eq. (40) is independent of the conformational properties of the chain, contrary to Eq. (39) (which depends on  $\nu$ ). Thus, it should be valid for both 2D and 3D dilute polymer solutions as well as for dense melts. While for the slithering-snake algorithm<sup>43</sup> and for some extended reptation algorithms<sup>51,76</sup> the scaling found for  $\tau_N$  is very close to Eq. (40), the behavior in dense systems is quite different.<sup>79,80</sup> The influence of density on the slithering-snake dynamics<sup>f</sup> has recently been studied by the bond-fluctuation model.<sup>80</sup> The following paragraphs briefly summarize some results of this work.

*From Dilute Solutions to Dense Melts: A Case Study by the BFM.* Reference 80 describes simulations for athermal systems containing chains of length  $16 \leq N \leq 1024$  at different volume fractions  $\phi$ .  $\phi$  ranges from dilute solutions to dense melts ( $\phi \approx 0.5$ ; see footnote on page 22). Figure 11 compares the relaxation time  $\tau_N$  in dilute solution with that in the melt. In dilute solution, the simulation results agree with the prediction of Eq. (40),  $\tau_N \sim N^2$ . This implies that the assumption of independent, free diffusive motion, which underlies Eq. (40), is well borne out. If this assumption was also true in the melt, the sole effect of density would be to slow down the monomer mobility  $m$ . However, the

<sup>e</sup>Here, we assume that the time to shift the monomer index along the chain is implemented in a way that it also takes a time of order 1 only.

<sup>f</sup>For local moves the set of allowed bond vectors automatically prevents bonds from crossing each other in the course of the simulation (see Sec. 4.1). If slithering-snake moves are considered, the uncrossability of the bonds has to be checked explicitly to avoid configurations which cannot be attained or unraveled by local updates. Bond crossing can occur if vectors from the classes  $[2, 2, 1]$  or  $[3, 1, 0]$  are selected.<sup>81</sup>

dependence of  $\tau_N$  on  $N$  should not change. Figure 11 shows that this is not true. At  $\phi = 0.5$ ,  $\tau_N$  increases exponentially with  $N$ . This strong slowing-down of the dynamics reflects correlations between the motion of the chains.

The importance of such intermolecular interactions for the polymer dynamics was first discussed by Deutsch.<sup>79</sup> However, Deutsch goes beyond a mere interpretation of the dynamic properties of the slithering-snake algorithm. He identifies the slithering dynamics with the physical dynamics along the primitive path in the reptation model (see Fig. 5). This suggests an attractive application: The slithering-snake algorithm mimics the back and forth reptation motion of real chains without modeling the (time consuming) local monomer fluctuations around the primitive path. It focuses on the long time behavior of very large chains, where all of these local motions have already relaxed. This suggests that the slithering-snake dynamics may be interpreted in terms of theories proposed for the dynamics of strongly entangled polymer melts, such as the one of Deutsch.<sup>79</sup>

The main results of this theory may be summarized as follows: A chain can reptate through the network of its neighbors only as long as the end monomer does not enter a dense region which prohibits any further forward move. The only way out of the trap is to partially retract and to explore the environment for new pathways. These intermolecular interactions create a free energy barrier which temporarily localizes the chain in the region it initially occupied, and protracts the relaxation. Further relaxation in a dense region could only occur if the chain end encounters another end which moves out of its way. This implies that the portion of the chain, which altered its initial configuration while exploring the environment, should span the typical distance between chain ends  $d_{\text{end}}$ . Let there be  $g$  monomers in this portion. Then, by exploiting the ideality of the chains in the melt, we have  $g = (d_{\text{end}}/b)^2 \sim N^{2/3}$  because the density of chain ends scales as  $\rho/N \propto d_{\text{end}}^{-3}$ . Thus,  $g$  is large for long chains. If we now assume that the monomers have to overcome the free energy barrier  $g\Delta\mu$ , where  $\Delta\mu$  is the difference in the monomer chemical potential between the newly explored environment and the region of the initial chain configuration, the barrier is large and the relaxation dynamics should be activated. Thus,  $\tau_N \propto N^2 \exp[\text{const } N^{2/3}]$ . This is the main prediction by Deutsch. The assumption of a finite  $\Delta\mu$  was challenged by Semenov<sup>82</sup> who suggested that the barrier is due to fluctuations of the molecular field rather than to a permanent chemical potential difference (see also Ref. 83). This picture implies that the barrier should be proportional to  $\sqrt{g}$  so that  $\tau_N \propto N^2 \exp[\text{const } N^{1/3}]$ .

The simulation data of Fig. 11 appear to agree with the latter prediction better than with the original one of Ref. 79 (at least for the chain lengths simulated up to now). Certainly, more work is needed to test these predictions.

*Slithering-Snake versus Local Moves.* From a merely computational point of view Fig. 11 appears to indicate that the slithering-snake dynamics is not very efficient in equilibrating dense melts. Its relaxation time increases with  $N$  more strongly than a power law which is typically found for local updating schemes.<sup>8</sup> Nevertheless, simulations of the BFM for short chains ( $N = 10, 20$ ) suggest that the slithering-snake algorithm decorrelates melt configurations ( $\phi \approx 0.5$ ) very efficiently.<sup>84,85</sup>

This point certainly needs more studies. Work in this direction was done in Ref. 80. Figure 12 shows a preliminary result for the diffusion coefficient  $D_N$  as a function of chain length.  $D_N$  was obtained from simulations employing a mixture of local and slithering-

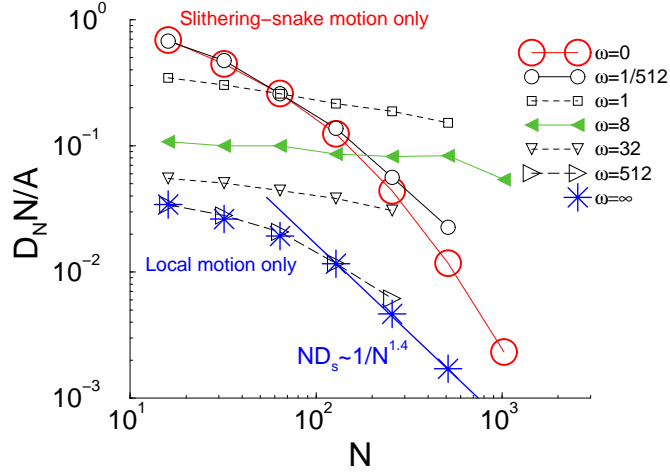


Figure 12. Spatial diffusion coefficient  $D_N$  versus  $N$  for the BFM at  $\phi = 0.5$ . Different ratios  $\omega$  of local to slithering-snake moves are compared:  $\omega = 0$  corresponds to pure slithering-snake dynamics,  $\omega = \infty$  to the pure local dynamics. The diffusion coefficient is scaled by  $N/A$ , where  $A$  is the acceptance rate. For both local and slithering-snake dynamics the acceptance rate is roughly  $A \approx 0.1$  for all  $N$  at  $\phi = 0.5$ . The data for  $\omega \ll 1$  and for  $\omega \gg 1$  are very similar to the pure slithering-snake ( $\omega = 0$ ) and the pure local limit ( $\omega = \infty$ ), respectively. For  $\omega \approx 8$ ,  $ND_N/A$  is approximately independent of  $N$ . This may define a reasonable choice of  $\omega$  for efficient equilibration of longer chains by local and slithering-snake moves. Adapted from Ref. 80.

snake moves. This introduces, as an additional parameter, the ratio  $\omega$  of local to slithering-snake moves. The figure indicates that pure slithering-snake dynamics ( $\omega = 0$ ) equilibrates short chains more efficiently than pure local dynamics ( $\omega = \infty$ ), in accord with the observations made in Refs. 84,85. By contrast, with increasing chain length  $D_N$  slows down exponentially for the slithering-snake algorithm, as expected due to  $D_N \sim R_g^2/\tau_N$ , whereas the local dynamics exhibits a crossover from Rouse-like,  $D_N \sim 1/N$ , to reptation-like behavior,  $D_N \sim 1/N^{\approx 2.4}$  (see Sec. 2.3). If this trend persists, the pure slithering-snake algorithm will become inefficient to equilibrate long chains. However, one can speculate that the addition of local moves weakens the confinement imposed by neighboring chains on the slithering-snake dynamics in the large- $N$  limit. Indeed, this seems to be borne out by the data. For short chains ( $N < 64$ )  $D_N$  decreases monotonously with increasing  $\omega$ , since local moves are less efficient in exploring the configuration space and the confinement is negligible. As  $N$  increases, one finds a non-monotonous behavior. The dynamics first becomes more rapid, as local moves are added. This effect appears to saturate at  $\omega \approx 10$ . Larger values of  $\omega$  causes the diffusion coefficient to decrease again strongly (at fixed  $N$ ). This implies that a judicious (model-dependent) choice of  $\omega$  is crucial if one wants to equilibrate a melt of long chains efficiently by mixing local and slithering-snake moves.

*Remark.* The efficiency of the slithering-snake algorithm (with or without local moves) or of the extended reptation algorithm deteriorates considerably as  $\phi$  approaches 1, since there is not sufficient space for the growth step. If one is interested in these high den-

sities,<sup>g</sup> an alternative simulation method may be provided by reptation moves including a “walker”.<sup>77,78</sup> A “walker” is defined as an isolated monomer (or as a small group of monomers). In the MC move, the “walker” attaches to a chain in its neighborhood, which then releases a monomer somewhere along its backbone, yielding a new “walker” at a different position than the original one. Since the walker can be created by cleaving a monomer from a chain, the algorithm works even at  $\phi = 1$ . In order to preserve monodispersity the update of the configuration is finished if the “walker” attaches again to the chain it was originally cleaved from.

### 5.3 Non-Local Moves: The Pivot Algorithm

A non-local  $N$ -conserving move attempts to update a chain portion of order  $N$ . If successful, it drastically modifies the chain dimension. Global properties, such as the end-to-end distance, should therefore relax within a few ( $N$ -independent) steps so that the critical slowing-down is largely moderated –if not removed.<sup>h</sup>

This appealing feature makes the search for appropriate non-local moves very attractive. However, not every conceivable move turns out to be efficient. This is mainly due to two reasons:

1. A drastic modification of the chain configuration is much more likely to violate the excluded-volume constraint than local or bilocal moves do. So, we expect the acceptance rate of non-local updates to decrease with  $N$ . The challenge consists in inventing moves whose acceptance rate does not rapidly vanish as  $N \rightarrow \infty$ .
2. A non-local move typically requires a CPU time of order  $N$  (checking self-avoidance, updating the configuration if accepted), in contrast to order 1 for a local or a bilocal move. The extra factor  $N$  must be compensated by a very efficient decorrelation of the configurations to justify the use of non-local moves.

A paradigm for a non-local algorithm, which satisfies these criteria, is the “pivot algorithm”.<sup>i</sup>

*Pivot Algorithm for the SAW.* The elementary move of the pivot algorithm works as follows (Fig. 13): First, one randomly chooses a monomer  $i$  and a symmetry operation (e.g. a rotation, a reflection, etc.). The monomer serves as a “pivot point” for the symmetry operation which turns the chain portion comprising the monomers  $i + 1, \dots, N$  to a new position, while the other piece of the chain (monomers  $1, \dots, i$ ) remains unchanged. The

<sup>g</sup>Note that  $\phi \approx 1$  means that the polymer melt has zero compressibility  $\kappa_T$ . By contrast, real polymer melts are compressible, with  $k_B T \rho \kappa_T$  being of the order  $10^{-1}$  for temperatures above the glass transition or crystallization temperatures.<sup>86</sup> For the BFM<sup>24</sup> at  $\phi \approx 0.5$ ,  $k_B T \rho \kappa_T \approx 0.2$  and for the bead-spring models of Sec. 4.2 at  $\rho \approx 0.9$ ,  $10^{-2} < k_B T \rho \kappa_T < 10^{-1}$  (see Refs. 24, 87). Thus, it appears that the limit  $\phi \rightarrow 1$  is not needed to model dense polymer melts.

<sup>h</sup>This idea is related to that of cluster algorithms<sup>35,88</sup> employed in MC simulations of spin systems near criticality. Close to the phase transition, the spins are strongly correlated. They form cluster of size  $\xi$  (= correlation length, corresponding to  $R_e$  in the polymer problem, see Sec. 1). A cluster algorithm finds one cluster (Wolff algorithm) or all of them (Swendsen-Wang algorithm) and updates all spins of the cluster(s) at once. This strongly reduces or even eliminates, in favorable cases, the critical slowing-down.<sup>35,88</sup>

<sup>i</sup>The pivot algorithm was invented by Lal in 1969. A comprehensive discussion of the algorithm may be found in Refs. 34, 89.

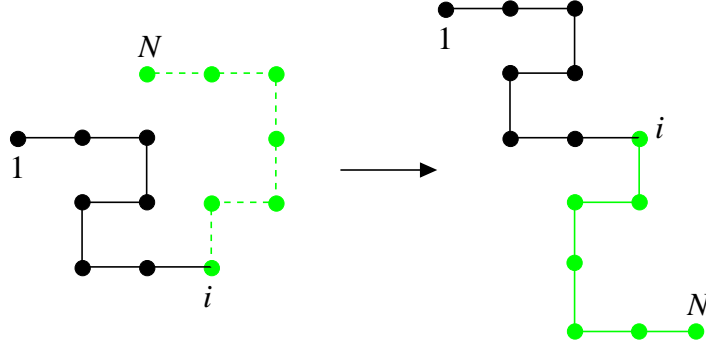


Figure 13. Illustration of the pivot algorithm. A monomer  $i$  (= pivot point) is chosen at random. It divides the chain into two pieces: The monomers  $1, \dots, i$  remain fixed, while the monomers  $i+1, \dots, N$  are translated to a new position via a randomly chosen symmetry operation (rotation, reflection, etc.). In the example of the figure, a  $180^\circ$  rotation around the monomer  $i$  is shown.

proposed move is accepted if the resulting configuration is self-avoiding. Otherwise, it is rejected and the old configuration is recounted.

Qualitatively, the pivot move resembles an attempt to construct a SAW of  $\sim N$  monomers by joining two SAW's of  $\sim N/2$  monomers at the pivot point. The probability for the result to be self-avoiding should scale as  $\mathcal{Z}(N)/\mathcal{Z}^2(N/2) \sim N^{-(\gamma-1)}$  [see Eq. (8)]. This heuristic argument suggests that the acceptance rate vanishes as  $N^{\approx-0.344}$  in 2D ( $\gamma = 43/32$ )<sup>21</sup> and as  $N^{\approx-0.158}$  in 3D ( $\gamma \simeq 1.158$ )<sup>21</sup>. Although these estimates are quantitatively not very accurate, they correctly predict the qualitative trends: The acceptance rate decreases with increasing chain length as a power law  $N^{-\alpha}$  and the exponent  $\alpha$  is larger in 2D than in 3D (2D:  $\alpha \approx 0.19$ , 3D:  $\alpha \approx 0.11$ ).<sup>89</sup>

*Relaxation Time and Computational Complexity.* Fortunately, the numerical value of  $\alpha$  is small, implying that even for long chains, e.g.  $N = 10^5$ , every  $N^{0.11} \approx 3.5$ th move is accepted. Since a successful move implies a huge modification of the conformation, one can expect global properties to relax after a few steps. So, the relaxation time  $\tau_N$  scales as  $\tau_N \sim N^\alpha$ . This increase is distinctly slower than that of all algorithms discussed so far. Due to this property and due to the fact that the pivot algorithm is known to be ergodic<sup>89</sup> it has become very popular (see e.g. the compilation of references in Ref. 90). Currently, the pivot algorithm is considered to be the most efficient algorithm for studying configurational properties<sup>j</sup> of isolated SAW's.<sup>21,89,92</sup>

The pivot algorithm quickly decorrelates global quantities, such as the end-to-end distance. However, it is not as efficient for local properties: The conformation of a specific monomer is only altered if this monomer is selected as a pivot point and if the move is successful. A successful moves takes a time of order  $N^\alpha$  and, as the chain consists of  $N$  monomers, the decorrelation time of a local observable should scale as  $\tau_{\text{loc}} \sim N^{1+\alpha}$ . This

<sup>j</sup>By configurational properties we mean quantities characterizing the chain dimension, such as  $R_e$ ,  $R_g$ , etc. A very accurate estimate of the critical exponent  $\nu$  in 3D ( $\nu = 0.5877 \pm 0.0006$ ) was obtained by the pivot algorithm.<sup>91</sup> By contrast, it appears difficult to measure precisely the partition function  $\mathcal{Z}(N)$ , and so the exponent  $\gamma$  with the pivot algorithm. For this purpose, other algorithms, such as the “join-and-cut” algorithm<sup>21</sup> or chain-growth algorithms<sup>92</sup>, are better suited. The current best estimate for  $\gamma$  in 3D is  $\gamma = 0.1575 \pm 0.0006$ .<sup>21</sup>

extra factor  $N$  is felt if one starts from an arbitrary initial configuration. Full equilibration on all length scales is required before large-scale equilibrium properties may be sampled. The equilibration time must be longer than the longest relaxation time in the system, i.e., than  $\tau_{\text{loc}}$ .

For a non-local algorithm the computational complexity is a particularly important quantity because inefficient implementations may ruin the advantage gained by fast decorrelation. The most naive check for self-avoidance would take a time of order  $N^2$  so that  $\tau_{\text{cc}} = N^2 \tau_N \sim N^{2+\alpha}$ , comparable to the slithering-snake algorithm [Eq. (40)]. Obviously, a faster check is called for. In Ref. 89 it was argued that, by starting at the pivot point and working outwards, self-intersections may be detected in a time of order  $N^{1-\alpha}$ . This procedure must be repeated  $\sim N^\alpha$  times to obtain one accepted pivot. So, the time required per accepted pivot scales as  $\sim N$ .<sup>k</sup> Once the pivot is accepted, we still have to update the monomer positions which also takes a time of order  $N$ . So, in total we find  $\tau_{\text{cc}} \sim N \tau_N \sim N^{1+\alpha}$  for global properties and  $\tau_{\text{cc}}^{\text{loc}} \sim N \tau_{\text{loc}} \sim N^{2+\alpha}$  for local properties. The estimate for  $\tau_{\text{cc}}^{\text{loc}}$  is again comparable to the slithering-snake algorithm [Eq. (40)]. Therefore, one could also use slithering-snake moves to engender an initial, equilibrated configuration for the pivot algorithm.

#### 5.4 Non-Local Moves in the Melt: The Double-Pivot Algorithm

Due to its efficiency in decorrelating SAW configurations it is tempting to apply the pivot algorithm also to other situations, such as the collapse transition, SAW's in confined geometry, or dense polymer melts. However, in these cases, the algorithm becomes inefficient. The non-local moves either lead to large energy differences (collapse transition) or violate the excluded-volume condition (SAW's in confined geometry, dense melts) so that they are rejected.

Should one therefore give up the idea of using pivot-like moves, say, for dense melts? Recent work suggests that this conclusion might be wrong. Instead of pivoting a piece of one chain to a new position the MC move can involve two chains. Such a move was termed “double-pivot (DP)” algorithm.<sup>l</sup> The basic idea of the algorithm is to cleave simultaneously a bond in a chain and in one of its neighbor chains, and to reconnect the monomers such that the chains remain monodisperse.<sup>m</sup> The algorithm works as follows (Fig. 14):

1. A monomer, say monomer  $i$  in chain  $a$ , is chosen at random. Around this monomer the neighborhood is inspected to find bridgeable neighbors on other chains  $b$ . A bridgeable neighbor is defined by the following requirements:
  - (a) It must be possible to connect the neighbor to  $i$  by a bond vector. This imposes a restriction on the intermolecular distance between  $i$  and its neighbor. In the ex-

<sup>k</sup>The pivot algorithm may be implemented so that the time required to obtain an accepted move is of order  $N^q$  with  $q < 1$ .<sup>90</sup>

<sup>l</sup>The double-pivot algorithm has been proposed recently in Ref. 93. In part, this work was motivated by a novel chain-bridging algorithm which was successfully employed in atomistic simulations of long polyethylene chains (see Ref. 94). Our presentation of the DP algorithm is inspired by the discussion of Ref. 94. Due to the newness of the algorithm the implementation that we propose might turn out not to be the most efficient one.

<sup>m</sup>In general, connectivity-altering moves between arbitrary monomer pairs of neighboring chains lead to a distribution of chain lengths, i.e., to polydispersity. The width of the distribution can be controlled by introducing a chemical potential. See Sec. 3.3 of Ref. 43 for further discussion in the context of lattice models and e.g. Ref. 94 for an application to atomistic MC simulations of polyethylene.

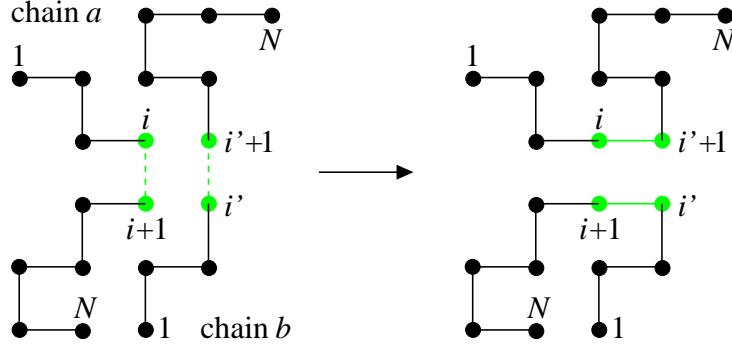


Figure 14. Illustration of a double-pivot (DP) move on a square lattice. A DP move flips bonds between two adjacent chains  $a$  and  $b$ . To this end, the neighborhood of monomer  $i$  in chain  $a$  is inspected to find potential bridging sites on chain  $b$ . To preserve monodispersity a potential bridging site has to satisfy the following conditions: (a) The  $(i + 1)$ th monomer of chain  $b$ , denoted by  $i' + 1$  in the figure, must be separated by a distance of the bond length from monomer  $i$ . (b) The same condition must also hold for the distance of the  $i$ th monomer of chain  $b$ , denoted by  $i'$ , from monomer  $i + 1$  of chain  $a$ . Between these four monomers a connectivity-altering move is attempted. The bonds from  $i$  to  $i + 1$  and from  $i'$  to  $i' + 1$  are broken, and new bonds,  $i$  with  $i' + 1$  and  $i'$  with  $i + 1$ , are created. The proposed move is accepted or rejected according to Eq. (43). From the figure it is clear that the DP algorithm can only be carried out if there are matching monomers on a neighbor chain within the distance of a bond. This is the more likely, the higher the concentration of the solution. Therefore, the algorithm (presumably) works best in concentrated solutions or melts. If successful, the move entails a drastic change of the chain configuration.

ample of Fig. 14 it must coincide with the lattice constant. For the BFM it must be among the set of allowed bond vectors [Eq. (33)], whereas, for a continuum model, the bond energy resulting from taking the intermolecular distance as a bond vector should not be so large that the proposed bond would never occur in equilibrium. In the latter case, it might be necessary to reduce the strength of the bond potential, e.g., the force constant of Eq. (34).<sup>93</sup>

- (b) To maintain monodispersity the neighbor must be either the  $(i - 1)$ th or the  $(i + 1)$ th monomer of the chain  $b$ . We distinguish the monomers of chain  $b$  by a “prime”, e.g.  $i'$ , from those of chain  $a$ .
- (c) If it is monomer  $i' \pm 1$ , monomer  $i'$  must be separated from monomer  $i \pm 1$  of chain  $a$  by a distance which satisfies condition (a).

Using these three criteria we determine the total number of bridgeable neighbors of monomer  $i$ ,  $N_{\text{DP}}(i, \mathbf{x}')$ . If  $N_{\text{DP}}(i, \mathbf{x}') = 0$  for all  $i$ , the configuration  $\mathbf{x}'$  must be updated by local (or bilocal) moves to bring the monomers in more favorable positions for bridging.<sup>93, 94</sup>

2. A double-pivot move is initiated by randomly selecting one of the bridgeable neighbors, say  $i' + 1$  in chain  $b$ , from  $N_{\text{DP}}(i, \mathbf{x}')$ . Then, the bonds between  $i$  and  $i + 1$  and between  $i'$  and  $i' + 1$  are cleaved, and one attempts to create new bonds between  $i$  and  $i' + 1$  and between  $i'$  and  $i + 1$ . This move just switches the connectivity while



preserving the chain length, and is proposed with probability

$$P_{\text{pro}}(\mathbf{x}' \rightarrow \mathbf{x}) = \frac{1}{N_{\text{DP}}(i, \mathbf{x}')} . \quad (41)$$

3. To satisfy detailed balance we have to determine  $P_{\text{pro}}(\mathbf{x} \rightarrow \mathbf{x}')$  of the reverse move. As the forward move only alters the connectivity between the chains, but does not displace the monomers, the number of bridgeable neighbors of a specific monomer remains unchanged. That is,  $N_{\text{DP}}(i, \mathbf{x}) = N_{\text{DP}}(i, \mathbf{x}')$ . To reverse the forward move, we have to select monomer  $i' + 1$  on chain  $b$  and its bridgeable neighbor  $i$  on chain  $a$ . This occurs with probability

$$P_{\text{pro}}(\mathbf{x} \rightarrow \mathbf{x}') = \frac{1}{N_{\text{DP}}(i' + 1, \mathbf{x}')} \quad (42)$$

so that the Metropolis criterion reads

$$\text{acc}(\mathbf{x}' \rightarrow \mathbf{x}) = \min \left( 1, \frac{N_{\text{DP}}(i, \mathbf{x}')}{N_{\text{DP}}(i' + 1, \mathbf{x}')} e^{-\beta[U(\mathbf{x}) - U(\mathbf{x}')] } \right) . \quad (43)$$

The difference  $U(\mathbf{x}) - U(\mathbf{x}')$  is the local change in energy due to the switching of the bonds between chains  $a$  and  $b$ .

The steps 1.–3. may be repeated several times. However, the number of iterations should not be too large. Otherwise it is likely that an accepted move annihilates one of its predecessors by performing the transition between two chains in the reverse direction. To avoid this inefficiency it is important to mix up the local configuration of the system. This may be achieved by e.g. local MC moves or by combining<sup>93</sup> the DP algorithm with MD simulations.

## 6 Monte Carlo Methods for Polymers: Rosenbluth Sampling and Its Modern Variants

The first MC method to simulate a SAW was “simple sampling”.<sup>43</sup> This static method (Sec. 3) works as follows:

1. Place the first monomer at the origin, randomly choose a bond vector, and append it to the monomer.
2. Choose the next bond vector, again randomly, connect it to the second monomer, and check the self-avoidance (Fig. 15).
3. If the chain is self-avoiding, the random growth process may be continued. If not, the self-avoiding piece of the SAW, obtained up to this point, must be discarded, and we have to start from scratch at the first step again.
4. The steps 1. to 3. are repeated until a SAW of the desired length  $N$  is obtained. Then, data analysis may be done.
5. Repeat steps 1. to 4. to gather sufficient statistics.

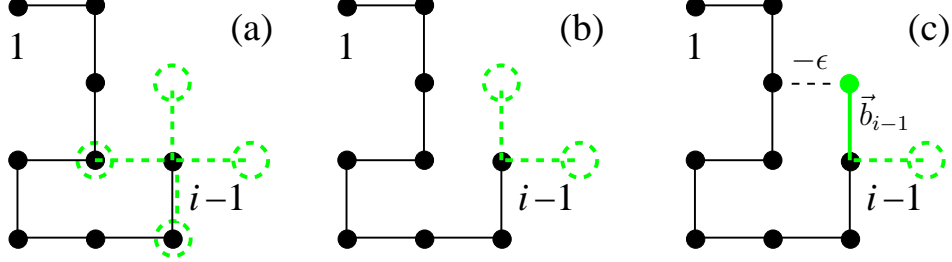


Figure 15. Illustration of the simple-sampling (a) and the Rosenbluth-Rosenbluth (RR) methods [(b) and (c)] on a square lattice. The coordination number of the lattice  $z (= 4)$  defines the number of possible bond vectors  $\{\vec{b}\}$ . In simple sampling, all of these vectors have the same a priori probability. Thus, the bond vector  $\vec{b}_{i-1}$  ( $= \vec{r}_i - \vec{r}_{i-1}$ ) from monomer  $i-1$  to the new monomer  $i$  can also point in the direction of already occupied lattice sites. This leads to the attrition problem. The RR method strongly reduces the attrition by taking  $\vec{b}_{i-1}$  only from the open directions  $k_i (= 2$  in the example of the figure). The new bond vector can be chosen either uniformly from the  $k_i$  possibilities, as indicated by the dashed lines and dashed circles in panel (b), or according to the local Boltzmann factor [Eq. (48)], if e.g. an attractive monomer-monomer interaction is present [filled grey circle in panel (c)].

Apparently, completed SAW's are independent of one another and occur with the same probability  $P_s(\mathbf{x}) = P_{\text{saw}}(N)$ . This is the main advantage of a static MC method. The main disadvantage of simple sampling is that  $P_{\text{saw}}(N)$  becomes exponentially small for large  $N$ . To see that, let us calculate  $P_{\text{saw}}(N)$  for a SAW on a hypercubic lattice. The hypercubic lattice has the coordination number  $z = 2d$ . Thus, the number of random walks (RW's) starting at the origin and having  $N-1$  steps, is  $\mathcal{Z}_{\text{RW}} = z^{N-1}$ .  $\mathcal{Z}_{\text{RW}}$  is the partition function of the RW. Out of these  $z^{N-1}$  random-walk configurations simple sampling selects those which are self-avoiding. As there are  $\mathcal{Z} [\sim \mu^N N^{\gamma-1}$ , Eq. (8)] such configurations,  $P_{\text{saw}}(N)$  is given by

$$P_{\text{saw}}(N) \sim \left(\frac{\mu}{z}\right)^N N^{\gamma-1} = N^{\gamma-1} e^{-\lambda N} \sim e^{-\lambda N} \quad (N \text{ large}), \quad (44)$$

where  $\lambda = \ln(z/\mu)$  is called “attrition constant”. The attrition constant is  $\lambda > 0$  (in 2D and 3D),<sup>n</sup> reflecting that the monomer partition function, i.e., the number of ways to place a monomer on the lattice, of a RW ( $= z$ ) is, due to the neglect of self-avoidance, larger than that of a SAW ( $= \mu$ ).

Equation (44) illustrates that simple sampling is not an efficient simulation method for SAW's. On average,  $e^{\lambda N}$  random walks have to be constructed to obtain one SAW (“attrition problem”). As  $\lambda = 0.416$  and  $0.248$  for the 2D and 3D hypercubic lattices, this number of constructions is prohibitively large already for  $N \gtrsim 50$ . Even if one modifies the construction method by avoiding immediate backfolds, thereby replacing  $z$  by  $z-1$  (“Non-Reversal Random Walk (NRRW)”), the exponential attrition remains. Generation of SAW's with  $N > 10^2$  is still unfeasible. Therefore, all alternative simulation techniques have to alleviate this attrition problem.

<sup>n</sup>For a compilation of attrition constants see Refs. 34, 43. If  $d \rightarrow \infty$ ,  $\lambda$  goes to zero as  $\lambda \rightarrow 1/2d$  for the hypercubic lattice.

## 6.1 Inversely Restricted Sampling: Rosenbluth-Rosenbluth Method

The attrition problem arises because simple sampling chooses blindly from the nearest-neighbor sites to place a new monomer. A more clever algorithm could scan the local environment around the last monomer and exclude those trial directions which lead to self-intersections. The position of the next monomer  $i$  can then be chosen with equal probability from the remaining  $k_i$  open directions (Fig. 15).<sup>o</sup> This method, known as “inversely restricted sampling” or “Rosenbluth–Rosenbluth (RR) algorithm”,<sup>p</sup> strongly reduces the attrition<sup>q</sup> at the expense of introducing a bias. A SAW is not generated with uniform probability, but with probability

$$P_s(\mathbf{x}) = \prod_{i=1}^N \frac{1}{k_i} \quad (k_1 = 1) . \quad (45)$$

Equation (45) shows that configurations with small  $k_i$ ’s have a higher probability of occurring. This bias toward dense configurations in the production of a SAW must be corrected in its analysis by the weight  $W(\mathbf{x}) \propto 1/P_s(\mathbf{x})$  when calculating observables [see Eq. (18)].

The RR algorithm is a static MC method. As such, it has the advantage that successively generated SAW’s are independent of each other. All problems of decorrelating configurations, discussed in Sec. 5, are absent by construction. On the other hand, Eq. (45) also points to the major difficulty of the method. The RR method favors dense configurations which are not representative of long SAW’s. Thus,  $P_s(\mathbf{x})$  differs from  $P_{eq}(\mathbf{x})$ . As  $N$  increases, the difference becomes more pronounced. To compensate the discrepancy between the two probabilities the distribution of weights must become broad: Dense configurations have small weights and open chains have in general larger weights. A detailed analysis of this problem was undertaken by Batoulis and Kremer.<sup>95</sup> They showed that the distribution of weights, obtained from  $M$  repetitions of the RR method ( $\{W(\mathbf{x}_m), m = 1, \dots, M\}$ ), is dominated by few configurations having the largest weights. The most relevant SAW configurations have, however, smaller weights. To sample this portion of the weight distribution sufficiently,  $M$  has to become very large [see the discussion of Eq. (22)]. This problem makes the RR algorithm not suitable for the simulation of long SAW’s.<sup>r</sup>

However, the RR algorithm should be well suited if the bias introduced by the sampling engenders configurations which are close to the physical ones. That is, if the equilibrium configurations are less swollen than those of a SAW. As this is the case close to  $\Theta$ -point in 3D (see Sec. 1), one might expect the RR algorithm to be more efficient for  $T \approx T_\Theta$ . In fact, this expectation is nicely borne out. The biased sampling of the RR method produces

<sup>o</sup>Here, we assume  $0 < k_i \leq z - 1$ . If all neighbor sites are blocked ( $k_i = 0$ ), the configuration obtained until then is trapped. It must be discarded, and the construction resumes from the beginning. The first monomer can be placed anywhere on the lattice. If the construction always starts, say, from the origin, we set  $k_1 = 1$  in Eq. (45).

<sup>p</sup>See Ref. 95 for a detailed statistical analysis of the RR algorithm<sup>96</sup> in the context of simulating SAW’s. Chapter 11 of Ref. 31 gives a discussion in the larger context of free energy calculations with applications to discrete and continuous chain models.

<sup>q</sup>In the RR algorithm, attrition occurs because the growing chain may be trapped, i.e.,  $k_i = 0$  for some  $i$ . In practice, this does not appear to be a serious problem in 3D for high-coordination lattices.<sup>95,97</sup> It was suggested that effects of trapping should become visible only for  $N \gtrsim 10^4$ .<sup>97</sup> This implies that there is still exponential attrition, as  $N \rightarrow \infty$ , but at a much lower rate than in Eq. (44).

<sup>r</sup>In the simulations on the (high-coordination) FCC lattice of Ref. 95 the systematic error due to the weights rendered a precise determination of  $R_g$  impossible for  $N \gtrsim 200$ .

an effective attraction between the monomers which closely resembles that of  $\Theta$ -chains. An important consequence of this special property is that the weights are nearly compensated by the Boltzmann factor.<sup>97</sup> Thus, the RR method was employed to study properties near the  $\Theta$ -point (see e.g. Refs. 10,11 of Ref. 97), and it also represents the core of a modern algorithm, the “Pruned-Enriched Rosenbluth Method (PERM)<sup>47</sup>”.

## 6.2 Pruned-Enriched Rosenbluth Method (PERM)

The simulation of a polymer chain close to the  $\Theta$ -point requires the introduction of an attraction between the monomers to compensate their mutual repulsion. Typically, these thermal interactions are modeled by a short range inter-monomer potential.<sup>s</sup> As a chain is grown according to the Rosenbluth scheme, the presence of the potential implies that the internal energy of a chain changes:

$$u_i(\vec{b}_{i-1}) = U(\vec{r}_1, \dots, \vec{r}_{i-1} + \vec{b}_{i-1}) - U(\vec{r}_1, \dots, \vec{r}_{i-1}) \quad [u_1(\vec{b}_0) := U(\vec{r}_1)] . \quad (46)$$

Here,  $U(\vec{r}_1, \dots, \vec{r}_{i-1})$  is the potential energy of chain having  $(i-1)$  monomers and  $\vec{b}_{i-1}$  is the bond vector from the  $(i-1)$ th to the  $i$ th monomer. A priori, there are different ways to incorporate  $u_i(\vec{b}_{i-1})$  in the construction. Two possible choices are the following:<sup>47</sup>

1. The first method is the classical Rosenbluth scheme. The position of the  $i$ th monomer is chosen from the free neighbors with uniform probability [see Eq. (45)]. Thus, the weight of the new chain configuration  $\mathbf{x} (= \vec{r}_1, \dots, \vec{r}_N)$  is given by<sup>t</sup>

$$W(\mathbf{x}) = \frac{e^{-\beta U(\mathbf{x})}}{P_s(\mathbf{x})} = \prod_{i=1}^N \left[ k_i e^{-\beta u_i(\vec{b}_{i-1})} \right] . \quad (47)$$

2. An alternative consists in including the Boltzmann factor in the probability  $P_{s,i}$  for placing the  $i$ th monomer. Let  $\{\vec{b}\}$  denote the ensemble of possible bond vectors. For the hypercubic lattice,  $\{\vec{b}\}$  coincides with the number of lattice directions  $z$ , for the BFM it is given by Eq. (33). Then, we may write for  $P_{s,i}$  (Fig. 15)

$$P_{s,i} = \frac{e^{-\beta u_i(\vec{b}_{i-1})}}{\sum_{\{\vec{b}\}} e^{-\beta u_i(\vec{b})}} = \frac{e^{-\beta u_i(\vec{b}_{i-1})}}{w_i} \xrightarrow{\text{SAW}} \frac{1}{k_i} , \quad (48)$$

where the normalization  $w_i$  reduces to  $k_i$  if only excluded-volume interactions are taken into account (SAW limit). This implies that the weight  $W(\mathbf{x})$  is given by

$$W(\mathbf{x}) = \frac{e^{-\beta U(\mathbf{x})}}{P_s(\mathbf{x})} = \prod_{i=1}^N \left[ \frac{e^{-\beta u_i(\vec{b}_{i-1})}}{P_{s,i}} \right] = \prod_{i=1}^N w_i . \quad (49)$$

<sup>s</sup>For the SAW on the hypercubic lattice the attraction is usually implemented between non-bonded nearest neighbors.<sup>47</sup> In a simulation of the  $\Theta$ -transition with the BFM, a square-well potential of range  $\sqrt{6}$  (in units of the lattice constant) was used.<sup>60</sup> This choice ensures that the first peak of the pair-distribution function is encompassed by the range of the potential. A more complicated choice was made in Ref. 98. In the continuum, a Gaussian chain model with a non-truncated LJ-interaction was extensively studied (see Refs. 64,99 and references therein).

<sup>t</sup>Note that, contrary to the definition of  $W(\mathbf{x})$  in Sec. 3, Eq. (47) does not include the factor of the (unknown) partition function.

Both methods have been used to simulate  $\Theta$ -polymers via the Rosenbluth algorithm.<sup>47</sup>

Despite the fact that, precisely at  $T_\Theta$ , the RR configurations more or less coincide with the equilibrium configurations, the accuracy of the method deteriorates for  $N > N_c$ .<sup>u</sup> Again, the reason is that  $P_s(x)$  does not perfectly agree with  $P_{eq}(x)$ . As a result, the weight distribution becomes so broad that chains with the biggest weights dominate the sample (see Fig. 2 of Ref. 47). Due to Eq. (18) this leads to a large variance of the computed observables.

To improve the accuracy one has to reduce the variance. Grassberger proposed a clever way to achieve this.<sup>47</sup> Assume that we have constructed a chain up to monomer  $i$  ( $1 \leq i \leq N$ ) via the RR method. This chain has the weight  $W_i$  which we want to prevent from fluctuating too much. That is, if  $W_i$  exceeds a lower or an upper bound, we interfere in the following way:

1. If  $W_i < W_i^-$ , we “prune” the sample: If  $W_i$  decreases below the threshold  $W_i^-$ , a random number  $0 \leq \zeta \leq 1$  is uniformly drawn. If  $\zeta < 1/2$ , the chain is removed. Otherwise, it is kept, its weight is doubled ( $W_i \rightarrow 2W_i$ ), and the step-by-step growth continues.
2. If  $W_i > W_i^+$ , we “enrich” the sample: If  $W_i$  exceeds the upper bound  $W_i^+$ ,  $c$  copies, typically 2,<sup>100</sup> of the configuration are made, each of which is given the new weight  $W_i \rightarrow W_i/c$ . These copies are then grown independently of each other.

This control of the weight distribution within the RR algorithm was termed “Pruned-Enriched Rosenbluth Method (PERM)”.<sup>47</sup>

Of course, the question arises of how to choose the bounds  $W_i^\pm$ . Here, it is important to note that neither the pruning nor the enrichment step introduces any bias. In the calculation of the sums in Eq. (18) the increase of  $W_i$  by pruning is compensated by the probability  $1/2$  with which the configuration is retained, and the decrease of  $W_i$  in the enrichment is compensated by the number of copies  $c$ . Thus, we are free to choose the bounds  $W_i^\pm$ . Bad choices can “only” render the method inefficient, but not incorrect. In order to determine optimum values for  $W_i^\pm$  the following procedure was proposed (for temperatures that are not too low):<sup>47,100</sup>

- First, one chooses  $W_i^- = 0$  and  $W_i^+$  very large. That is, one performs a simulation via the original RR method. This simulation yields the weights  $W_i$  for  $i = 1, \dots, N$ . First estimates for the bounds  $W_i^\pm$  are then determined by  $W_i^- = C^- W_i$  and  $W_i^+ = C^+ W_i$  with  $C^+/C^- \approx O(1)$ - $O(10)$ .
- These estimates are refined “on the fly”. Imagine that we have obtained  $M_i$  configurations of chain length  $i$  from the simulation. Then, we first calculate the partition

<sup>u</sup>The critical chain length  $N_c$  depends on the model. For a simple cubic lattice it is  $N_c \approx 10^3$ .<sup>47</sup>

<sup>v</sup>Enrichment is a classical technique for simulating SAW's. Briefly, it works as follows: If a chain survives the  $s$  step,  $c$  copies of its configuration are made, which serve as independent starting points for further growth. The method may be implemented in a “breadth-first” or a “depth-first” fashion. The former implies that all copies are first grown to size  $2s$  before the entire sample is copied again. By contrast, the latter method tries to complete the construction of one copy up to chain length  $N$  before passing to next one. The pros and cons of the two implementations are discussed in the context of the PERM in Ref. 47. More details about enrichment may be found in Refs. 34, 43.

function by  $[\mathbf{x} = (\vec{r}_1, \dots, \vec{r}_i)$ , see Eq. (18)]

$$\mathcal{Z}_i = \int d\mathbf{x} e^{-\beta U_i(\mathbf{x})} \approx \frac{1}{M_i} \sum_{m=1}^{M_i} \frac{e^{-\beta U_i(\mathbf{x}_m)}}{P_s(\mathbf{x}_m)} = \frac{1}{M_i} \sum_{m=1}^{M_i} W_i(\mathbf{x}_m), \quad (50)$$

and from that, we determine the new bounds by  $W_i^\pm = C^\pm \mathcal{Z}_i$ .

*Applications of the PERM.* The PERM was invented to simulate the transition from an excluded-volume to a collapsed chain at the  $\Theta$ -temperature  $T_\Theta$ . Theoretically, the transition is usually identified with a tricritical point in the limit  $N \rightarrow \infty$ .<sup>16</sup> A tricritical point exhibits mean-field behavior in 3D. Thus, one expects that  $\nu = 1/2$  and  $\gamma = 1$  [Eq. (8)] at  $T_\Theta$  for  $N \rightarrow \infty$ . This asymptotic large- $N$  behavior is supplemented by (universal) corrections of order  $1/\ln N$  for finite chain length (see e.g. Chap. 21 of Ref. 16 for a good discussion). A significant test of the theory therefore requires to study very long chains.

In Ref. 47 such a test was attempted. Grassberger performed a comparative study of various models employed in the literature: a SAW on a simple cubic lattice with attractive nearest-neighbor interactions, the BFM with two versions for the attractive monomer-monomer interactions,<sup>60,98</sup> and a LJ bead-spring model<sup>64</sup>. Using the PERM these models could be simulated with high precision and, partly, with much longer chain lengths than studied before. These simulations yielded refined estimates of  $T_\Theta$  for the various models, confirmed the mean-field-like asymptotic character of the  $\Theta$ -point, but also showed that the leading-order logarithmic corrections cannot explain the finite- $N$  behavior found, even for  $N = 10^4$ .<sup>w</sup>

On the technical side, it was found that the selection of the position of the next monomer  $i$  with uniform probability from the  $k_i$  open directions is only the best choice for the SAW on the simple cubic lattice (first method of page 36). As alluded to at the end of Sec. 6.1, this is due to a near cancellation of the Rosenbluth weight and the Boltzmann factor: Many nearest-neighbor contacts lead to a low Rosenbluth weight, but to a large Boltzmann factor, and vice versa. For more long-range or more complicated interactions the degree of cancellation need not necessarily be the same. In fact, for the BFM it was found in Ref. 47 that a selection of the next monomer position according to the Boltzmann factor [Eq. (48)] is more efficient. Similar approaches were also used to study e.g. simple models of proteins.

The preceding discussion appears to suggest that the PERM is a single-chain technique. This is not true. We just quote two recent examples. The PERM was utilized to simulate the denaturation transition of a simple model for double-stranded DNA (two SAW's).<sup>102</sup> A truly multi-chain system was studied in Ref. 103. This work is concerned with the phase diagram of semidilute polymer solutions for  $T \leq \Theta$  (see Fig. 3). For a review of these and other applications see Ref. 100.

<sup>w</sup>These findings elicited further theoretical<sup>101</sup> and numerical work.<sup>99,101</sup> On the theoretical side, subleading corrections of order  $\ln(\ln N)/(\ln N)^2$  were calculated and found to be as large as the leading  $1/\ln N$  term. On the numerical side, MC simulations<sup>101</sup> for  $N \leq 10^4$  of a NRRW, including weakly attractive two-body, but repulsive three-body interactions, were performed. This model shows logarithmic corrections which are much weaker than those found in Ref. 47 and are roughly compatible with the theoretical predictions. However, Ref. 101 stresses a problem in the analysis of the  $\Theta$ -point. To estimate  $T_\Theta$ , an infinite- $N$  property, precisely from the simulations one has to rely on the theoretical predictions for the finite- $N$  corrections to extrapolate to  $N \rightarrow \infty$ . To this end, the simulated chains must be long enough for these corrections to apply. This regime appears to be very hard to attain, even for  $N \sim 10^6$ .

### 6.3 Configurational-Bias Monte Carlo and Recoil-Growth Algorithm

An alternative multi-chain MC scheme, incorporating the RR method, is the Configurational-Bias Monte Carlo (CBMC) algorithm.<sup>x</sup> Contrary to the PERM, CBMC builds up a new chain step-by-step without controlling the weights  $w_i$  “on the go”. It is only after a successful construction that the resultant bias is removed: The new chain is accepted according to the Metropolis criterion with a probability dependent on the total Rosenbluth weights of the new and the old chain configurations. This additional test warrants sampling from the Boltzmann distribution.

A recent extension of CBMC is the “Recoil-Growth (RG) algorithm”.<sup>y</sup> Contrary to the RR method, which only looks ahead one step while constructing a chain, the RG algorithm uses a more sophisticated growth procedure. It places a long retractable feeler at the head of a growing chain. The feeler spys out the environment to find favorable pathways for the chain construction. The efficiency of the method resides in the fact that the growth does not terminate if the feeler encounters a trap. It merely recoils back from the trap and pursues its search in a different direction. After the construction is completed, the new chain replaces the old one, just as in CBMC, with a probability determined by their respective weights according to the Metropolis criterion.

#### 6.3.1 Configurational-Bias Monte Carlo (CBMC)

To illustrate the CBMC scheme in more detail we consider a solution of SAW’s on a lattice (Fig. 16):

1. Given the initial configuration  $\mathbf{x}'$  of the system we randomly select a chain and one of its monomers. Let this be the monomer  $n'$  of chain  $b$  ( $1 \leq n' \leq N$ ). The configuration of the chain portion  $i = n', \dots, N$  is characterized by the sequence of bonds  $\vec{b}_{n'-1}, \dots, \vec{b}_{N-1}$ . Each bond represents a specific choice from the set of all possible bond vectors  $\{\vec{b}\}$ . So, we can write the Rosenbluth weight of the monomers  $n', \dots, N$  of chain  $b$  as

$$W^b(\mathbf{x}') = \prod_{i=n'}^N w_i^b(\mathbf{x}') , \quad w_i^b(\mathbf{x}') = e^{-\beta u_i^b(\vec{b}_{i-1})} + \sum_{\vec{b} \neq \vec{b}_{i-1}} e^{-\beta u_i^b(\vec{b})} . \quad (51)$$

Here,  $u_{n'}^b(\vec{b}_{n'-1})$  is the energy of monomer  $n'$  at its actual position in the chain. It includes the interactions with the monomers of all other chains and with the monomers  $i = 1, \dots, n' - 1$  of its own chain [Eq. (46)]. The monomers  $n' + 1$  to  $N$  have to be omitted because one thinks of the chain  $b$  as being (re-) constructed step-by-step via the RR procedure with probability [Eq.(48)]

$$P_{s,i}^b(\mathbf{x}') = \frac{e^{-\beta u_i^b(\vec{b}_{i-1})}}{w_i^b(\mathbf{x}')} . \quad (52)$$

<sup>x</sup>CBMC was introduced by Siepmann and Frenkel in the 1990s. It can be applied to lattice and off-lattice models. The initial off-lattice applications have demonstrated the power of the algorithm for the study of a large variety of problems in polymer physics. Therefore, CBMC has become an important and widely used simulation technique. A comprehensive and very pedagogical account of the method, including flowcharts of the algorithm and examples, is given in the textbook by Frenkel and Smit.<sup>31</sup>

<sup>y</sup>The RG algorithm was introduced in Refs. 104, 105. A detailed description may be found in Chap. 13.7 of the textbook by Frenkel and Smit.<sup>31</sup> For practical applications, it is very helpful that the FORTRAN codes of the “Case Studies” may be downloaded from [http://molsim.chem.uva.nl/frenkel\\_smit](http://molsim.chem.uva.nl/frenkel_smit).

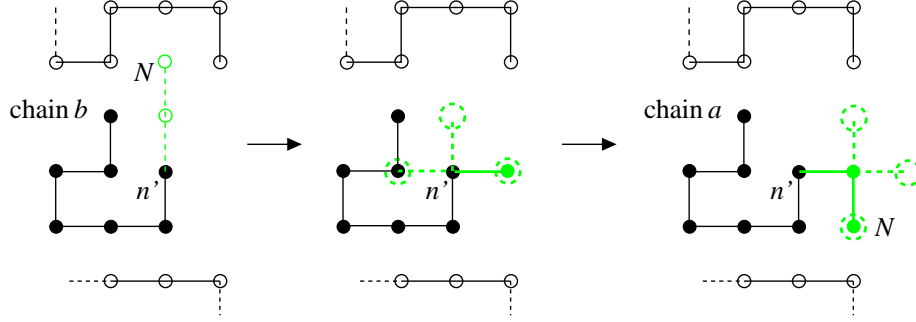


Figure 16. Schematic of a CBMC move executed in a polymer solution. The initial configuration  $\mathbf{x}'$  of the solution is shown in the left figure. From all chains of  $\mathbf{x}'$  the chain  $b$  and one of its monomers,  $n'$ , are chosen. Here,  $n' = N - 2$ . The chain portion  $n', \dots, N$  is deleted (dashed grey lines connecting the open grey circles) and reconstructed step-by-step. Upon completion we obtain a new chain configuration, “chain  $a$ ”, and so a new configuration,  $\mathbf{x}$ , of the solution. In the construction, the new bond  $\vec{b}_{i-1}$  from monomer  $i - 1$  to monomer  $i$  ( $i = N - 1, N$ ) is selected from the set of bond vectors  $\{\vec{b}\}$  according to the local Boltzmann factor  $P_{s,i}^a(\mathbf{x})$  [Eq. (55)]. For lattice models this set, which defines the number of trial directions at each step, is finite and closely related to lattice structure: E.g. for the hypercubic lattice, it coincides with the number of lattice directions  $z$  (or  $z - 1$  if backfolds shall be excluded a priori). For the BFM the trial directions may be taken as the ensemble of allowed bonds [Eq. (33)]. For a continuum model there is a priori an infinite number of possible directions, from which a suitable finite number of trial directions must be selected so that new chain configurations are efficiently generated (see Chap. 13.3 of Ref. 31 for details). If only excluded volume interactions are present, as assumed in the figure,  $P_{s,i}^a(\mathbf{x}) = 1/2$  for  $i = N - 1$  and  $P_{s,i}^a(\mathbf{x}) = 1/3$  for the last monomer  $N$ . Thus, the total weight  $W^a(\mathbf{x})$  of chain  $a$  is 6, whereas that of the chain portion  $n' = N - 1, N$  of the old chain  $b$  is  $W^b(\mathbf{x}') = 4$ . Thus,  $W^a(\mathbf{x})/W^b(\mathbf{x}') = 1.5$ , and chain  $a$  will be accepted according to Eq. (58).

Thus, the potential energy of the chain portion  $n', \dots, N$  is given by

$$U^b(\mathbf{x}') = \sum_{i=n'}^N u_i^b(\vec{b}_{i-1}), \quad (53)$$

and the probability to propose the chain portion may be expressed as

$$P_s^b(\mathbf{x}') = \prod_{i=n'}^N P_{s,i}^b(\mathbf{x}') = \frac{e^{-\beta U^b(\mathbf{x}')}}{W^b(\mathbf{x}')} . \quad (54)$$

2. The monomers  $n'$  to  $N$  of chain  $b$  are deleted. This corresponds to a “shrinkage-growth” implementation of the algorithm, which is (presumably) more efficient than a “growth-shrinkage” procedure (see Sec. 5.2).
3. To obtain a new configuration  $\mathbf{x}$  of the system the chain  $b$  is fully ( $n' = 1$ ) or only in part ( $n' > 1$ ) reconstructed:
  - If  $n' = 1$ , we start building a new chain by randomly placing the first monomer somewhere in system. Let this new chain be labeled “chain  $a$ ”. At the position, where the monomer is inserted, it interacts with the other chains of the system. It has an energy  $u_{n=1}^a(\vec{b}_0)$  [ $= U(\vec{r}_1)$ , see Eq. (46)], giving rise to the weight  $w_1 = \exp[-\beta u_1^a(\vec{b}_0)]$ .



- If  $n' > 1$ , we rebuild the chain portion  $i = n', \dots, N$  monomer-by-monomer via the RR method. This growing chain, comprising the initial portion of chain  $b$  ( $i = 1, \dots, n' - 1$ ) and the newly attached monomers ( $i = n, \dots, N$ ) will also be called “chain  $a$ ”. In the following we suppress the prime ( $'$ ) to indicate that a new chain conformation is obtained, even if the first  $n-1$  monomers are identical with those of chain  $b$ . To add the  $n$ th monomer we proceed as described in the next step.
4. A monomer may be attached to the growing chain via a bond from the set of possible bond vectors  $\{\vec{b}\}$ . Out of these trial directions we choose one according to Eq. (48). This implies for the monomer  $i$  ( $= n, \dots, N$ )

$$P_{s,i}^a(\mathbf{x}) = \frac{e^{-\beta u_i^a(\vec{b}_{i-1})}}{\sum_{\{\vec{b}\}} e^{-\beta u_i^a(\vec{b})}} = \frac{e^{-\beta u_i^a(\vec{b}_{i-1})}}{w_i^a(\mathbf{x})}, \quad (55)$$

where  $u_i^a(\vec{b}_{i-1})$  is the change in potential energy of the system due to the addition of the new bond  $\vec{b}_{i-1}$  from monomer  $i-1$  to monomer  $i$  [Eq. (46)].

5. The preceding step is repeated until the construction of the chain is completed. Thus, the new chain configuration occurs with probability [Eq. (49)]

$$P_s^a(\mathbf{x}) = \prod_{i=n}^N P_{s,i}^a(\mathbf{x}) = \frac{e^{-\beta U^a(\mathbf{x})}}{W^a(\mathbf{x})}, \quad (56)$$

where  $W^a(\mathbf{x})$  and  $U^a(\mathbf{x})$  are defined analogously to Eqs. (51,53).

6. Now, we recognize that  $P_s^a(\mathbf{x})$  may be interpreted as the probability of proposing a transition from the old configuration  $\mathbf{x}'$  to the new configuration  $\mathbf{x}$ . Correspondingly,  $P_s^b(\mathbf{x}')$  is the probability for the reverse step. That is,

$$P_s^a(\mathbf{x}) = P_{\text{pro}}(\mathbf{x}' \rightarrow \mathbf{x}) \quad \text{and} \quad P_s^b(\mathbf{x}') = P_{\text{pro}}(\mathbf{x} \rightarrow \mathbf{x}'). \quad (57)$$

Finally, we insert Eq. (57) into Eq. (29) to obtain the acceptance probability for the new configuration

$$\begin{aligned} \text{acc}(\mathbf{x}' \rightarrow \mathbf{x}) &= \min \left( 1, \frac{P_{\text{pro}}(\mathbf{x} \rightarrow \mathbf{x}')}{P_{\text{pro}}(\mathbf{x}' \rightarrow \mathbf{x})} e^{-\beta[U^a(\mathbf{x}) - U^b(\mathbf{x}')] } \right) \\ &= \min \left( 1, \frac{W^a(\mathbf{x})}{W^b(\mathbf{x}')} \right). \end{aligned} \quad (58)$$

*Discussion.* An important feature of the CBMC method is its non-local character: A successful CBMC construction implies a large-scale configurational change. Either a new chain is inserted somewhere in the system –this may be employed very efficiently to study phase equilibria of polymer solutions in the bulk<sup>106</sup> and in thin films<sup>107</sup>– or part<sup>z</sup> of a chain

<sup>z</sup>This part need not necessarily start at monomer  $n' + 1$  and terminate at the chain end, as assumed in our previous discussion. It can also comprise an inner portion of the chain, say the monomers  $n' + 1, \dots, n' + m' - 1 < N$ . In this case, the reconstruction has to satisfy the additional condition that it starts at monomer  $n'$  and ends at the position of monomer  $n' + m'$  (see Chap. 13.4 of Ref. 31). This variant of CBMC may be used to relax e.g. ring polymers, which have no free ends so that the method described above could not be applied.

is regrown. This leads to a rapid decorrelation of chain configurations and to efficient sampling, provided the system has a low or moderate density and the chains are not too long. When dealing with long chains and/or dense melts the following problems occur:

- Chain construction in CBMC is based on the Rosenbluth method, yielding a distribution of configurations that differs from the Boltzmann distribution. This difference becomes more pronounced with increasing  $N$ , even for an isolated chain at the  $\Theta$ -point (see Secs. 6.1,6.2). As CBMC does not control the weights while synthesizing the chain, contrary to the PERM, the acceptance rate for chain reconstructions falls exponentially in  $N$  for large chain lengths.
- Another problem results from the “shortsightedness” of the RR algorithm. By looking only one step ahead, the chain construction can run into traps. For isolated chains this trapping implies that there is still an exponential attrition in  $N$  for long chains, albeit with an attrition constant much smaller than  $\lambda$  (see Eq. (44) and footnote on page 35).
- In a dense system, in addition to trapping, a further problem occurs. If a (part of a) new chain is inserted, it is fairly likely to be constructed just in the space originally occupied by the (part of the) old chain which was removed. This can lead to strong correlations between the new and old chain configurations so that sampling becomes inefficient.<sup>75</sup> In this situation, it might be best to combine CBMC with the slithering-snake algorithm (Sec. 5.2). That is, to try to regrow just the terminal bond at the other chain end according to Eq. (58). In a dense melt, it thus appears as if CBMC cannot be expected to decorrelate chain configurations more efficiently than the slithering-snake algorithm.

### 6.3.2 Recoil-Growth (RG) Algorithm

The RG algorithm was suggested as an alternative to CBMC,<sup>a</sup> exhibiting two major changes:

1. Instead of looking one step ahead the RG algorithm scans the environment via a retractable “feeler”. The feeler consists of a self-avoiding chain portion having at most  $N_{\text{recoil}}$  monomers.<sup>b</sup> The ability of the feeler to shrink and to grow helps to circumvent dense regions. This allows for the search of suitable pathways to complete the construction of the chain.

<sup>a</sup>The RG algorithm was introduced for SAW’s on a cubic lattice in Ref. 104. Reference 105 extends this study to continuum models. A comprehensive discussion of these works may be found in the textbook by Frenkel and Smit (Chap. 13.7).<sup>31</sup>

<sup>b</sup>Of course, in general  $N_{\text{recoil}} > 2$ , implying that the feeler is longer than one bond. Otherwise, the “shortsightedness” of CBMC is not removed. The idea to improve the RR method by looking several steps ahead is not new. It is embodied e.g. in the “scanning method” of Meirovitch.<sup>108</sup> This method still uses a one-step growth, but chooses a new bond  $\vec{b}_i$  according to the probability that a SAW of  $N_{\text{scan}}$  monomers can be constructed in direction of  $\vec{b}_i$ . As this implies an enumeration of all possible SAW’s of length  $N_{\text{scan}}$  starting at some monomer  $i$ , the scanning parameter  $N_{\text{scan}}$  is usually much smaller than  $N$ . Thus, trapping cannot be avoided completely. This would only be the case if  $N_{\text{scan}} = N - i$ , i.e., if one scanned all possible ways to complete the chain up to monomer  $N$  in direction of  $\vec{b}_i$ . For a comparative discussion of the RG algorithm and the scanning method see Ref. 104.

2. Contrary to CBMC, the incremental weights  $w_i$  for each newly added monomer are not calculated “on the fly”, but only after a new chain has been successfully constructed. Thus, the computation of the weights is carried out only once. In CBMC, it can happen that a lot of time is spent to calculate the weights of a partially grown chain which must then be discarded because the construction has run into a trap before completion.

*Description of the RG Algorithm.* These differences show that the RG algorithm comprises two independent steps: a construction step using the retractable feeler and an acceptance step including the weight calculation. We discuss these steps separately. In our discussion we assume that a whole chain is inserted in the system after a randomly chosen old chain has been removed (“shrinkage-growth procedure”). As before, the new chain is called “chain  $a$ ”, the old “chain  $b$ ”, and the respective new and old configurations of the system are denoted by  $\mathbf{x}$  and  $\mathbf{x}'$ .

Construction step (Fig. 17):

1. We place the first monomer of chain  $a$  at the random trial position  $\vec{r}_1$  and determine its energy  $U(\vec{r}_1) = u_1^a(\vec{b}_0)$ . To decide whether the position is accessible (“open”) we accept it with probability (see e.g. Ref. 31 for further discussion of this point)

$$P_1^{\text{open}}(\mathbf{x}) = \min \left( 1, e^{-\beta u_1^a(\vec{b}_0)} \right) \xrightarrow{\text{SAW}} \begin{cases} 1 & \text{if } \vec{r}_1 \text{ is unoccupied,} \\ 0 & \text{otherwise,} \end{cases} \quad (59)$$

where “SAW” means that there are only hard-core interactions.<sup>c</sup> If the position is “open”, we continue with step 2. Otherwise, step 1 is repeated.

2. Assume that the chain construction has arrived at monomer  $i$ . We randomly choose a new bond  $\vec{b}_i$  from monomer  $i$  to monomer  $i+1$ , compute  $P_{i+1}^{\text{open}}(\mathbf{x})$  for the trial position of  $i+1$ , and decide whether it is open or not according to Eq. (59). If it is closed, we keep choosing new bonds up to a maximum number of  $k_{\text{RG}}$  trial directions.<sup>d</sup> As soon as the first open direction is found, we proceed to step 3.

Otherwise, a recoil step is performed. That is, the chain moves back to monomer  $i-1$  where it searches for an open direction, using the  $k_{\text{rest}} = k_{\text{RG}} - k_{\text{checked}}$  previously unchecked directions. The first open direction found is used to place (again) monomer  $i$ . If the chain fails to find an open direction from the  $k_{\text{rest}}$  remaining ones at monomer  $i-1$ , it recoils to  $i-2$  and checks the previously unused directions for a possibility to grow from there. In difficult situations the chain keeps falling back until a maximum number of  $N_{\text{recoil}}$  recoil steps has been performed. If this number is exceeded, the construction of chain has to resume from step 1.

<sup>c</sup>In the SAW-limit, the decision of whether the position is open is not probabilistic, but deterministic. One just checks on overlaps with other monomers: If no overlap occurs, the position is open. For continuous potentials, however, the decision becomes probabilistic. We compare  $P_1^{\text{open}}(\mathbf{x})$  to a random number  $\zeta$ , uniformly distributed between 0 and 1. The position is open if  $\zeta < P_1^{\text{open}}(\mathbf{x})$ . Otherwise, it is closed. This means that we accept the position with probability  $P_1^{\text{open}}(\mathbf{x})$ .

<sup>d</sup>In practical implementations, the number of trial directions need not be the same for every monomer  $i$ . It may depend on  $i$ . For instance, one could choose for half of the monomers 2 trial directions and for the other half 3 directions so that  $k_{\text{RG}} = 2.5$  on average. Thus,  $k_{\text{RG}}$  need not be integer.

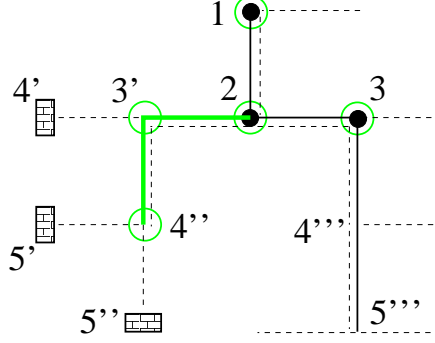


Figure 17. Illustration of the recoil-growth procedure for a SAW on a square lattice, using  $k_{\text{RG}} = 2$  and  $N_{\text{recoil}} = 3$ . The first monomer is placed at an empty (“open”) lattice site. At each site that the growing chain visits, it has  $k_{\text{RG}}$  trial directions for the next step (indicated by the dashed lines: there are 2 for each monomer). These directions are chosen at random from the  $z-1$  forward lattice directions. Starting at monomer 1 the chain first tries the path 1 2 3'. At that point, the first monomer ( $3 - N_{\text{recoil}} + 1$ ) becomes fixed (indicated by  $\circ \rightarrow \bullet$  in the figure). Then, the chain attempts to grow to position 4' and finds it blocked. Thus, it recoils to monomer 3' and tries the remaining direction leading to monomer 4''. At that point, the monomer  $4 - N_{\text{recoil}} + 1 = 2$  becomes fixed ( $\circ \rightarrow \bullet$ ), as the “feeler” 2 3' 4'' has attained its full length  $N_{\text{recoil}} = 3$  (grey thick solid line). However, the next two trial directions,  $4' \rightarrow 5'$  and  $4'' \rightarrow 5''$ , are found blocked. So, the feeler first recoils to 3', where it realizes that  $k_{\text{RG}}$  trial directions have been exhausted so that it must fall back its maximum length  $N_{\text{recoil}}$  to monomer 2. From there, it finds the open path 2 3 4''' 5''' so that monomer 3 becomes permanently attached to the chain. In this way, the construction goes on up to monomer  $N - N_{\text{recoil}} + 1$ . If this monomer was fixed, the construction stops, as a feeler to monomer  $N$  exists. Figure adapted from Ref. 104.

This shows that the RG algorithm is characterized by two parameters:  $k_{\text{RG}}$  and  $N_{\text{recoil}}$ . Tuning of these parameters is important to optimize the efficiency of the method.

3. If the construction has successfully placed the  $n$ th monomer, we attach monomer  $n - N_{\text{recoil}} + 1$  permanently to the chain, as recoiling can only fall back to  $n - N_{\text{recoil}} + 1$ . If this occurs and still no open direction is found at monomer  $n - N_{\text{recoil}} + 1$ , the growth process terminates and we must go back to step 1. Thus, the  $N_{\text{recoil}}$  monomers,  $i = n - N_{\text{recoil}} + 1, \dots, n$ , may be regarded as “retractable feeler” of maximum length  $N_{\text{recoil}}$  which probes the territory ahead of monomer  $n - N_{\text{recoil}}$ .
4. Repeat steps 2 and 3 until the monomer  $N - N_{\text{recoil}} + 1$  was attached to the chain. This implies that the feeler attained the chain end. A complete chain of length  $N$  has thus been constructed.

Acceptance step: If chain  $a$  has been successfully constructed, we have to determine its weight  $W^a(x)$  and that of the old chain  $b$ ,  $W^b(x')$ , which chain  $a$  attempts to replace. This may be done as follows:

1. In the construction step each monomer  $i$  disposes of (at most)  $k_{\text{RG}}$  trial directions into which the growth of a feeler of maximum length  $N_{\text{recoil}}$  may be attempted. Assume that we have checked  $k = 1, \dots, k_{\text{checked}}$  of these directions and found that the  $k_{\text{checked}}$ th direction is the first allowing us to complete the construction of the feeler

up to length  $N_{\text{recoil}}$ . We now test the remaining  $k_{\text{rest}} = k_{\text{RG}} - k_{\text{checked}}$  directions to find out how many “feelers” of length  $N_{\text{recoil}}$  can be grown. (For the last monomers,  $N - N_{\text{recoil}} + 1 < i \leq N$ , the length of the feeler is shortened by one bond, as  $i$  advances step-by-step toward  $N$ .) For monomer  $i$  let  $m_i(\mathbf{x})$  denote the number of directions where a feeler of full length can be grown. We have  $1 \leq m_i(\mathbf{x}) \leq 1 + k_{\text{rest}}$ . As monomer  $i$  is only irrevocably added to the chain, if its position was initially open [Eq. (59)] and if at least one feeler of length  $N_{\text{recoil}}$  may be grown from it, the probability with which we propose to place monomer  $i$  is given by [see Eq. (55)]

$$P_{s,i}^a(\mathbf{x}) = P_i^{\text{open}}(\mathbf{x}) \cdot \frac{1}{m_i(\mathbf{x})} = \frac{1}{w_i^a(\mathbf{x})}. \quad (60)$$

With respect to CBMC  $P_{s,i}^a(\mathbf{x})$  differs by the absence of the Boltzmann factor  $\exp[-\beta u_i^a(\vec{b}_{i-1})]$ , since only excluded-volume interactions are taken into account during the construction. So, the new chain  $a$  is proposed with probability

$$P_s^a(\mathbf{x}) = \prod_{i=1}^N \frac{1}{w_i^a(\mathbf{x})} = \frac{1}{W^a(\mathbf{x})} \quad (m_N(\mathbf{x}) = 1). \quad (61)$$

2. The weight calculation of the old chain  $b$  has two ingredients. First, we calculate  $P_i^{\text{open}}(\mathbf{x}')$  for each monomer  $i$  according to Eq. (59).<sup>e</sup> Second, we attempt to grow feelers in  $k_{\text{RG}} - 1$  randomly chosen directions and we count the number of feelers that attain length  $N_{\text{recoil}}$ . This number plus 1 yields  $m_i(\mathbf{x}')$  (“plus 1”, since one feeler of length  $N_{\text{recoil}}$  already exists along the backbone of chain  $b$ ). In analogy to chain  $a$ , we thus write

$$P_s^b(\mathbf{x}') = \prod_{i=1}^N \frac{P_i^{\text{open}}(\mathbf{x}')}{m_i(\mathbf{x}')} = \prod_{i=1}^N \frac{1}{w_i^b(\mathbf{x}')} = \frac{1}{W^b(\mathbf{x}')} \quad (m_N(\mathbf{x}') = 1). \quad (62)$$

3. Finally, the total potential energy of chains  $a$  and  $b$ ,  $U^a(\mathbf{x})$  and  $U^b(\mathbf{x}')$ , must be computed before the new chain can be accepted with probability [see Eqs. (57,58)]

$$\text{acc}(\mathbf{x}' \rightarrow \mathbf{x}) = \min \left( 1, \frac{W^a(\mathbf{x})}{W^b(\mathbf{x}')} e^{-\beta[U^a(\mathbf{x}) - U^b(\mathbf{x}')] } \right). \quad (63)$$

The RG algorithm has two adjustable parameters:  $k_{\text{RG}}$ , the (average) number of trial directions of a monomer, and  $N_{\text{recoil}}$ , the length of the retractable feeler. Intuitively, one expects that  $N_{\text{recoil}}$  should be large, whereas  $k_{\text{RG}}$  should be small. The main advantage of the RG algorithm is its ability to avoid traps by probing the environment with a feeler. A short feeler will strongly reduce this ability and thus the rate of successful chain construction. On the other hand, the value of  $k_{\text{RG}}$  should not be too large because  $1 \leq m_i(\mathbf{x}) \leq k_{\text{RG}}$ . (Remember that  $m_i(\mathbf{x})$  denotes the number of feelers of length  $N_{\text{recoil}}$ , which may be grown from monomer  $i$ .) A large  $k_{\text{RG}}$  allows for many different values of  $m_i(\mathbf{x})$ , as we go along the chain to calculate the weight  $W^a(\mathbf{x})$ . The spread in  $m_i(\mathbf{x})$  leads to a wide distribution of  $W^a(\mathbf{x})$ , which will strongly deviate from the Boltzmann distribution (see Sec. 6.1).

<sup>e</sup>For the old chain we only calculate  $P_i^{\text{open}}(\mathbf{x}')$ . This is different from the construction step, where we decide whether the position is open or closed according to probability  $P_i^{\text{open}}(\mathbf{x})$ . That is, we first calculate  $P_i^{\text{open}}(\mathbf{x})$  and then compare it to a random number  $\zeta$  uniformly distributed between 0 and 1. If  $\zeta < P_i^{\text{open}}(\mathbf{x})$ , the position is open. Otherwise, it is closed [see Eq. (59)].

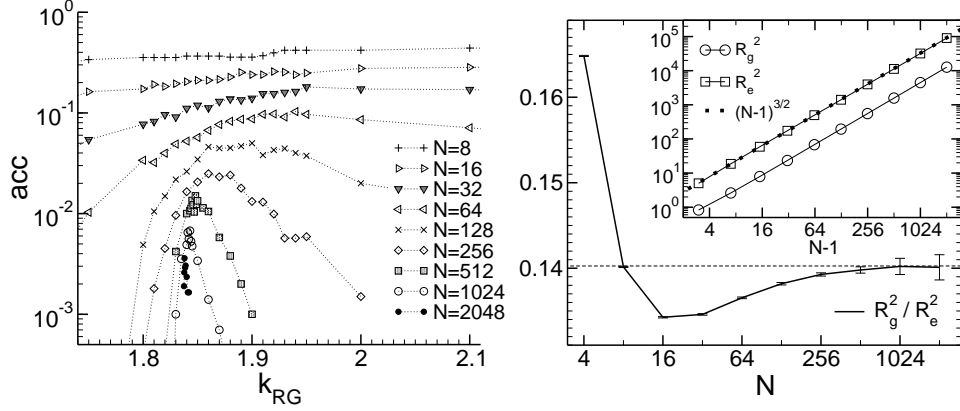


Figure 18. Left: Acceptance probability versus  $k_{RG}$  for  $8 \leq N \leq 2048$  obtained from simulations of isolated 2D chains (“Kremer-Grest model”) via the RG algorithm. For all cases,  $N_{\text{recoil}} = N/2$ . Note the sharp peak of  $\text{acc}(k_{RG})$  for long chains. The value of  $k_{RG}$  at the peak yields the optimum choice  $k_{RG}^{\text{opt}}(N)$  for this chain length. For  $k_{RG}^{\text{opt}}$  we find:  $\text{acc} \propto 1/N$ . Right: Plot of  $R_g^2/R_e^2$  versus  $N$  (main figure) and of  $R_e^2$  and  $R_g^2$  versus the number of bonds  $N-1$  (inset). The dashed horizontal line in the main figure represents the large- $N$  limit of  $R_g^2/R_e^2$  [Eq. (64)]. The dotted line in the inset shows the theoretically predicted power law  $R_e^2 \propto R_g^2 \sim N^{2\nu}$  with  $\nu = 3/4$ .

*Applications of the RG Algorithm.* These expectations are confirmed by applications of the RG algorithm to 3D lattice<sup>104</sup> and 3D off-lattice<sup>105</sup> models of polymer solutions. For the chain lengths studied ( $N \leq 100$ ) good choices are  $2 \lesssim k_{RG} \lesssim 3$  and  $3 \lesssim N_{\text{recoil}} \lesssim 10$ , with the need to have larger values for both parameters if the density of the solution increases. Furthermore, it was observed that, for high densities and long chains, the RG algorithm may be an order of magnitude more efficient than CBMC.

Our preliminary results<sup>109</sup> on 2D polymer solutions, simulated with the Kremer-Grest model (see Sec. 4.2), appear to confirm the trends observed in 3D. Presumably due to the fact that the risk of trapping is more severe in 2D than in 3D,<sup>95,110</sup> we found that, even for an isolated chain, chain lengths  $N \gtrsim 100$  are very difficult to simulate via CBMC. Here, the RG algorithm provides a powerful alternative. While the performance of the algorithm depends only weakly on the choice of  $N_{\text{recoil}}$ , provided a long feeler is employed, e.g.  $N_{\text{recoil}} = N/2$ ,  $k_{RG}$  must be optimized carefully. Figure 18 shows that the acceptance probability develops a pronounced peak for long chains. While for  $N \lesssim 64$  the acceptance probability is fairly insensitive to the precise choice of  $k_{RG}$ , if  $k_{RG} \approx 2$ , it has to be adjusted to 4 significant digits for  $N = 2048$ . In this case, the optimum value is  $k_{RG}^{\text{opt}} = 1.838$ . This means that on average there are four monomers with two trial directions, followed by one monomer with only one trial direction.<sup>f</sup>

To exemplify that the RG algorithm produces correct statistical properties Fig. 18 also shows the radius of gyration  $R_g^2$ , the end-to-end distance  $R_e^2$ , and the ratio  $R_g^2/R_e^2$ . For isolated, two-dimensional chains one expects to find the critical exponent  $\nu = 3/4$ <sup>22</sup> and

<sup>f</sup>We arrive at that conclusion in the following way. Suppose we allow the monomers to have either 1 or 2 trial directions. Let  $p$  denote the probability that a monomer has two trial directions. Then, we pose  $k_{RG} = 2 \cdot p + 1 \cdot (1 - p)$  so that  $p = 0.838$  with  $k_{RG}^{\text{opt}} = 1.838$  for  $N = 2048$ . Thus, out of 5 monomers roughly four monomers have two trial directions, leaving one trial direction for the remaining monomer.

the ratio<sup>111</sup>

$$\lim_{N \rightarrow \infty} \frac{R_g^2}{R_e^2} = 0.14026 \pm 0.00011 . \quad (64)$$

The figure illustrates that both predictions are well borne out by the simulation data. Apparently, the 2D Kremer-Grest model quickly converges to the large- $N$  limit, i.e, to  $R_e^2 \propto R_g^2 \sim N^{3/2}$  and to Eq. (64). Deviations are only visible when investigating the ratio  $R_g^2/R_e^2$  for  $N \lesssim 512$ . For longer chains corrections-to-scaling are small. In this respect, the Kremer-Grest model agrees with the results obtained for the SAW on a square lattice.<sup>111</sup>

## 7 Conclusions

Behind the title of this chapter “Monte Carlo Simulation of Polymers: Coarse-Grained Models” a topic of a large breadth is hidden. So, a selection is necessary. We have employed several criteria in this selection.

First, we concentrate on simple generic models retaining only basic features of a polymer chain (chain connectivity, excluded volume, etc.; see Sec. 4). Coarse-grained models derived from specific polymers are only touched upon briefly (Appendix A), although this is an important current research topic.<sup>41,42</sup>

Second, we define a generic model as one consisting of monomers with the simplest imaginable structure. They are identified with sites on a lattice or with Lennard-Jones spheres in the continuum. The monomers in the chain are all the same and uncharged (“neutral homopolymers”). Their interaction is either purely repulsive or consists of a short-range repulsion supplemented by an attractive potential at larger distances (see Sec. 4). Thus, nor did we consider specific interactions, such as electrostatic interactions, H-bonds, interactions between different species of monomers (e.g. binary mixtures, block-copolymers), etc. –this will be done in other chapters of this book– neither did we discuss current coarse-graining approaches which do not model a chain as a concatenation of monomers, but represent the whole polymer as a soft ellipsoidal<sup>112,113</sup> or spherical<sup>114</sup> particle.

Within the scope of these generic models we presented various algorithms. What is the upshot of this discussion for applications? Here are some suggestions:

- Dilute solutions:
  - For isolated chains, exempt of strong monomer-monomer interactions, the pivot algorithm is currently considered as being the most efficient method to study global properties related to the chain extension ( $R_g$ ,  $R_e$ , the exponent  $\nu$ , etc.). To study properties related to the partition function (e.g. the exponent  $\gamma$ ) other algorithms are better suited (see footnote on page 30).
  - To initialize the simulation via the pivot algorithm or for the analysis of local properties the slithering-snake algorithm represents an attractive alternative (see discussion on page 31).
  - If attractive monomer-monomer interactions are present, as it is the case close to the  $\Theta$ -point, it appears as if the pruned-enriched Rosenbluth method (PERM) is currently the most efficient algorithm (Sec. 6.2).

- From dilute solutions to dense melts:
  - As the density of the solution increases, the large-scale pivot moves become quickly inefficient. Here, either bilocal updating schemes, such as the slithering-snake algorithm, or non-local chain reconstructions via configurational-bias Monte Carlo (CBMC) are better suited. CBMC has become a well established tool, particularly in its grand-canonical formulation for the study of phase diagrams. It transpires that its range of applicability may be extended by the recently proposed recoil-growth (RG) algorithm (see Sec. 6.3). When the solution becomes more and more concentrated, the probability to renew the configuration of large chain portions becomes small. In this limit, the CBMC and RG methods reduce to the slithering-snake algorithm.
  - In dense melts consisting of long chain ( $N \gtrsim 10^3$ ) also the slithering-snake algorithm faces serious problems to equilibrate the system (Sec. 5.2). Here, it is better to use connectivity-altering moves instead of attempting to regrow (parts of) a chain. An example is the recently proposed double-pivot algorithm which appears to be very efficient in equilibrating dense melts (see Sec. 5.3).

The previous points only refer to the study of conformational and structural properties of polymer systems as well as to an efficient equilibration of the system. If the focus of interest are dynamic properties, local moves should be employed because they mimic best the physical dynamics (if solvent-mediated hydrodynamic interactions are absent; see Ref. 14).

The suggestions made above are a result of the discussion given in this chapter. Certainly, our discussion suffers from omissions. Personally, we feel that the most serious one are generalized ensemble techniques,<sup>115,116</sup> such as simulated or parallel tempering (see Chap. 14.1 of Ref. 31 for an introduction). In the context of polymer physics, these methods have been applied e.g. to determine the chemical potential of polymer chains (simulated tempering)<sup>117</sup>, to accelerate the equilibration of dense polymer melts (parallel tempering)<sup>118</sup> or to the simulation of phase transitions<sup>119</sup>.

We apologize for this review-like end of our report, but hope that its content, together with the bibliography, will be helpful for those interested in MC simulations of polymer models.

## Appendix

### A Realistic Models and Coarse-Graining of Real Polymers

The presentation of this chapter concentrated on generic coarse-grained models (Sec. 4). They are particularly useful for the study of the large-scale behavior of polymers. For these generic models several simulation algorithms were discussed. If one is now interested in properties of a specific molecule or material, the presented methods can still be applied. But their efficiency must be tested for the particular application.<sup>75</sup> The simulation of real materials must use models which reflect the underlying chemical structure of the polymer, even at the coarse-grained level. So, some mapping between a detailed, chemically realistic and a coarse-grained model must be established.<sup>41,42,120</sup> The following paragraphs explain



the need and the main steps of such a coarse-graining procedure and briefly describe an application to modeling a melt of poly(vinyl alcohol).

*From Realistic to Coarse-Grained Models.* An ideal simulation approach would consist in studying systems of long chains ( $N \gtrsim 10^3$ ) with potentials being calculated from the simultaneous motion of all nuclei and electrons (via the Car-Parrinello method<sup>121</sup>). However, already for an isolated chain such an approach is not feasible due to the large spread of subatomic and molecular time scales.<sup>4,5</sup> For  $N \sim 10^3$  the relaxation time of a polymer in dilute solution is about 1  $\mu$ s, whereas the inclusion of the electrons requires a time step of  $\sim 10^{-17}$  s in a MD simulation. This disparity of 11 orders of magnitude cannot be covered in present day simulations. A further restriction results from the system sizes that may be simulated. Typically, the total number of particles (electrons and nuclei) is limited to  $\sim 10^3$ .

Thus, simplifications are necessary.<sup>g</sup> There are several levels to this.<sup>120</sup> In a first step, the “on the fly” calculated quantum-mechanical potentials can be replaced by empirical potentials for the bond lengths, the bond angles, the torsional angles and the non-bonded interactions between distant monomers along the backbone of the chain (“quantum level  $\rightarrow$  atomistic level”). After careful optimization of the parameters of these potentials quantitative agreement between simulation and experimental results can be obtained.<sup>41,15</sup> If the system is to stay in thermal equilibrium on both the local monomeric scale and the global scale of the chain, such comparisons between simulation and experiments are limited to high temperatures up to now. Extensions to low temperatures where crystallization or sluggish glass-like relaxation occur still represent a great challenge for simulations on atomistic scale.<sup>h</sup> To tackle these problems the models must be further simplified. In a second step of simplification, fast degrees of freedom (bond length and bond angle vibrations, etc.) may therefore be eliminated by a coarse-graining procedure (“atomistic level  $\rightarrow$  coarse-grained level”). This procedure leads either to generic models, as those which were discussed in the previous sections, or to a coarse-grained model for a specific polymer (see Sec. 4). Recent approaches to the latter case have been reviewed in Refs. 41, 42, 120.

The first step, from the quantum to the atomistic level, may be interpreted as an ab-initio approach or a bottom-up construction of the model. When pushing the simplifications of the model beyond the atomistic to the coarse-grained level it becomes more and more apparent that the empirical potentials are just simulation parameters. So, instead of the bottom-up construction, one may also choose a more pragmatic top-down procedure. That is, given some (measured) properties, what are the potentials which can reproduce them?<sup>i</sup> In practice, one often proceeds as follows: By the bottom-up method a first guess of coarse-grained potentials is constructed. In a second step, these potentials are optimized with respect to some experimentally or theoretically known properties. With the final potentials

<sup>g</sup>Here, we assume that the whole chain is simulated at the same level of modeling. Of course, it is also possible to treat some part of the chain in atomistic detail (e.g. via Car-Parrinello), whereas others are modeled on a coarse-grained level. A recent example of such a multiscale approach is the selective adsorption of polycarbonate on a nickel surface.<sup>122</sup>

<sup>h</sup>An important contribution to this field are the “amorphous cell” simulations pioneered by Theodorou and Suter<sup>123,124</sup>. Here, the idea is to fold an infinitely long chain in the simulation box such that the monomer density is close to the experimental one and the resulting chain structure is reasonable. With this approach mechanical properties of the amorphous, glassy melt have been studied successfully.

<sup>i</sup>One such method for the solution of the inverse problem, known as “reverse Monte Carlo”, is actually used to interpret neutron and x-ray scattering data.<sup>125,126</sup>

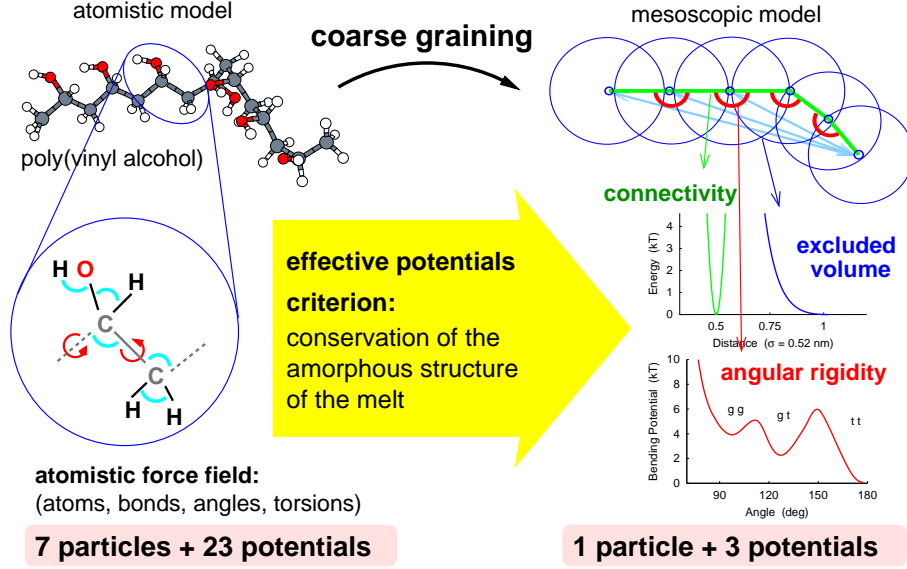


Figure 19. Schematic illustration of the coarse-graining procedure developed in Ref. 127. Starting from an atomistically detailed model of poly(vinyl alcohol) a bead-spring model is constructed. A monomer of the atomistic model is represented by a sphere on the coarse-grained level. The spheres interact by a harmonic bond potential and a purely repulsive LJ-like potential [Eq. (65)]. The parameters of these potentials are optimized in the following way. In an atomistic simulation at high temperature, the structure of a melt of poly(vinyl alcohol) and the local conformation of a polymer are determined: the structure by the pair-distribution function  $g(r)$ <sup>59</sup> and the conformation by the distribution of the angle between two vectors formed by connecting the C-atom of the CHOH-group of the  $(i - 1)$ th monomer with that of the  $i$ th monomer (first vector) and that of the  $i$ th monomer with that of the  $(i + 1)$ th monomer (second vector). From the angular distribution a potential of mean force<sup>59</sup> is calculated. In this case, it could directly be used as the bending potential for the bond angle in the coarse-grained model. Contrary to that, the parameters of the LJ-potentials are iteratively optimized until  $g(r)$  of the poly(vinyl alcohol) melt coincides with that of the coarse-grained model.

one may then (try to) predict quantities which are not easily measurable. In the following we will sketch one such pragmatic approach.

*An Example of How to Coarse-Grain Real Polymers.* The idea and the realization of the coarse-graining are illustrated by the example of poly(vinyl alcohol) in Fig. A.<sup>127</sup>

On the coarse-grained level, the chemical structure of the monomers is suppressed. They are represented by spheres, bound to each other by a harmonic potential. The orientation of successive bonds along the chain is determined by a bond angle potential, whereas other non-bonded monomers interact by the repulsive part of a LJ-like potential

$$U_{\text{LJ}}^{\text{rep}}(r) = \begin{cases} \epsilon_n(\sigma/r)^n - \epsilon_6(\sigma/r)^6 + \epsilon_0 & \text{for } r \leq r_{\min} , \\ 0 & \text{else ,} \end{cases} \quad (65)$$

where  $r_{\min}$  is the minimum of the potential and  $\epsilon_0$  is chosen such that the minimum is shifted to zero to have a continuous force. When fitting the potential to real polymers it

turned out<sup>67</sup> that for the repulsive term, an exponent smaller than 12 may be better adapted because the coarse-grained units have to be somewhat softer than atoms.<sup>j</sup> By comparing high-temperature simulations of an atomistically detailed and a coarse-grained model, the potential parameters of the latter are adjusted in such a way that it faithfully reproduces the backbone rigidity of poly(vinyl alcohol) and the local packing of its monomers in the liquid state.

The construction of the model was done at one state point. It is a priori not guaranteed how far from this reference conditions the model still matches the behavior of the original one. It turns out that in the present case, the crystallization at lower temperatures is well reproduced.<sup>129</sup>

## Acknowledgments

The results reported here, as far as it concerns our own work, were obtained in fruitful collaboration with M. Aichele, J.-L. Barrat, O. Biermann, K. Binder, R. Faller, B. Lobe, E. Luitjen, M. Müller, F. Müller-Plathe, L. Mattioni, W. Paul, D. Reith, and M. Wolfgardt. The authors' research is supported in part by the ESF Supernet programme and by the LEA "Macromolecules in nanoscopically structured media". J.B. is indebted to the IUF for financial support. Furthermore, we gratefully acknowledge generous grants of computing time by the IDRIS (Orsay, France).

## References

1. T. P. Lodge and M. Muthukumar, *Physical Chemistry of Polymers: Entropy, Interactions, and Dynamics*, J. Phys. Chem. **100**, 13275 (1996).
2. H. M. Jaeger, S. R. Nagel, and R. P. Behringer, *Granular solids, liquids, and gases*, Rev. Mod. Phys. **68**, 1259 (1996).
3. R. A. L. Jones, *Soft Condensed Matter* (Oxford University Press, New York, 2002).
4. K. Binder, *Introduction: General Aspects of Computer Simulation Techniques and their Applications in Polymer Science*. In: *Monte Carlo and Molecular Dynamics Simulations in Polymer Science* (Oxford University Press, New York, 1995), pp. 3–46.
5. K. Kremer, *Computer Simulations for Macromolecular Science*, Macromol. Chem. Phys. **204**, 257 (2003).
6. P. G. de Gennes, *Scaling Concepts in Polymer Physics* (Cornell University Press, Ithaca, 1996).
7. M. Doi and S. F. Edwards, *The Theory of Polymer Dynamics* (Oxford University Press, Oxford, 1986).
8. K. Binder and W. Paul, *Monte Carlo Simulations of Polymer Dynamics: Recent Advances*, J. Poly. Sci. B **35**, 1 (1997).
9. T. C. B. McLeish, *Tube theory of entangled polymer dynamics*, Adv. Phys. **51**, 1379 (2002).

<sup>j</sup>By doing more radical coarse-graining steps lumping many monomers<sup>128</sup> or even whole chains into one particle<sup>114</sup>, one will end up with soft overlapping potentials, as used e.g. in "Dissipative Particle Dynamics" (see Ref. 14).

10. M. Allen, *Introduction to molecular dynamics simulation*. In: *Computational Soft Matter: From Synthetic Polymers to Proteins* (NIC Series, 2004), pp. 1–28.
11. M. Müller, *Phase behavior and chain conformations in polymer blends: Monte Carlo simulation vs mean field theory*. In: *Computational Soft Matter: From Synthetic Polymers to Proteins* (NIC Series, 2004), pp. 225–264.
12. C. Holm, *Efficient methods for long range interactions in periodic geometries plus one application*. In: *Computational Soft Matter: From Synthetic Polymers to Proteins* (NIC Series, 2004), pp. 155–196.
13. A. Mark and S.-J. Marrink, *Self-organization of biomolecular systems: Simulating the folding and aggregation of peptides, proteins and lipids*. In: *Computational Soft Matter: From Synthetic Polymers to Proteins* (NIC Series, 2004), pp. 325–326.
14. B. Dünweg, *Advanced simulations for hydrodynamic problems: Lattice Boltzmann and dissipative particle dynamics*. In: *Computational Soft Matter: From Synthetic Polymers to Proteins* (NIC Series, 2004), pp. 61–82.
15. G. D. Smith, W. Paul, M. Monkenbusch, and D. Richter, *A comparison of neutron scattering studies and computer simulations of polymer melts*, Chem. Phys. **261**, 61 (2000).
16. L. Schäfer, *Excluded Volume Effects in Polymer Solutions* (Springer, Berlin–Heidelberg, 1999).
17. P. J. Flory, *Statistical Mechanics of Chain Molecules* (Wiley, New York, 1969).
18. J.-Ph. Bouchaud and A. Georges, *Anomalous Diffusion in Disordered Media: Statistical Mechanisms, Models and Physical Applications*, Phys. Rep. **195**, 127 (1990).
19. M. E. Fisher, *The renormalization group in the theory of critical behavior*, Rev. Mod. Phys. **46**, 597 (1974).
20. C. Domb, *The Critical Point* (Taylor & Francis, London, 1996).
21. S. Caracciolo, M. S. Causo, and A. Pelissetto, *High-precision determination of the critical exponent  $\gamma$  for self-avoiding walks*, Phys. Rev. E **57**, R1215 (1998).
22. A. Pelissetto and E. Vicari, *Critical phenomena and renormalization-group theory*, Phys. Rep. **368**, 549 (2002).
23. A. Y. Grosberg and A. R. Khokhlov, *Statistical Physics of Macromolecules* (AIP Press, New York, 1994).
24. K. Binder, J. Baschnagel, and W. Paul, *Glass Transition of Polymer Melts: Test of Theoretical Concepts by Computer Simulation*, Progr. Poly. Sci. **28**, 115 (2003).
25. J.-U. Sommer and G. Reiter (Eds.), *Polymer Crystallization: Observations, Concepts and Interpretations*, Lecture Notes in Physics **606** (Springer, Berlin–Heidelberg, 2003).
26. S. F. Edwards, *The size of a polymer molecule in a strong solution*, J. Phys. A: Math. Gen. **8**, 1670 (1975).
27. W. Paul, K. Binder, D. W. Heermann, and K. Kremer, *Crossover scaling in semidilute polymer solutions: a Monte Carlo test*, J. Phys. II **1**, 37 (1991); *Dynamics of polymer solutions and melts. Reptation predictions and scaling of relaxation times*, J. Chem. Phys. **95**, 7726 (1991).
28. S.-H. Chong and M. Fuchs, *Mode-Coupling Theory for Structural and Conformational Dynamics of Polymer Melts*, Phys. Rev. Lett. **88**, 185702 (2002).
29. T. P. Lodge, N. A. Rotstein, and S. Prager, *Dynamics of Entangled Polymer Liquids: Do Linear Chains Reptate?* Adv. Chem. Phys. **79**, 1 (1990).

30. H. Tao, T. P. Lodge, and E. D. von Meerwall, *Diffusivity and Viscosity of Concentrated Hydrogenated Polybutadiene Solutions*, *Macromolecules* **33**, 1747 (2000).
31. D. Frenkel and B. Smit, *Understanding Molecular Simulation – From Algorithms to Applications* (Academic Press, San Diego, 2nd ed. 2002).
32. K. Binder and D. W. Heermann, *Monte Carlo Simulation in Statistical Physics* (Springer, Berlin–Heidelberg, 2002).
33. D. P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, Cambridge, 2000).
34. A. D. Sokal, *Monte Carlo Methods for the Self-Avoiding Walk*. In: *Monte Carlo and Molecular Dynamics Simulations in Polymer Science* (Oxford University Press, New York, 1995), pp. 47–124.
35. A. D. Sokal, *Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms*. In: *Quantum Fields on the Computer* (World Scientific, Singapore, 1992).
36. D. Frenkel, *Introduction to Monte Carlo methods*. In: *Computational Soft Matter: From Synthetic Polymers to Proteins* (NIC Series, 2004), pp. 29–60.
37. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. N. Teller, and E. Teller, *Equation of state calculations by fast computing machines*, *J. Chem. Phys.* **21**, 1087 (1953).
38. O. Narayan and A. P. Young, *Convergence of Monte Carlo simulations to equilibrium*, *Phys. Rev. E* **64**, 021104 (2001).
39. W. Feller, *An Introduction to Probability Theory and its Applications*, Vol. 1 and Vol. 2 (Wiley, New York, 1966).
40. V. I. Manousiouthakis and M. W. Deem, *Strict detailed balance is unnecessary in Monte Carlo simulation*, *J. Chem. Phys.* **110**, 2753 (1999).
41. F. Müller-Plathe, *Coarse-Graining in Polymer Simulation: From the Atomistic to the Mesoscopic Scale and Back*, *ChemPhysChem* **3**, 754 (2002).
42. J. Baschnagel, K. Binder, P. Doruker, A. A. Gusev, O. Hahn, K. Kremer, W. L. Matice, F. Müller-Plathe, M. Murat, W. Paul, S. Santos, U. W. Suter, V. Tries, *Bridging the Gap Between Atomistic and Coarse-Grained Models of Polymers: Status and Perspectives*, *Adv. Polym. Sci.* **152**, 41 (2000).
43. K. Kremer and K. Binder, *Monte Carlo Simulations of Lattice Models for Macromolecules*, *Comp. Phys. Rep.* **7**, 259 (1988).
44. K. Binder, *Monte Carlo and Molecular Dynamics Simulations of Amorphous Polymers*. In: *Computational Modeling of Polymers* (Marcel Dekker, New York, 1992), pp. 221–295.
45. H.-P. Wittmann and K. Kremer, *Vectorized version of the bond fluctuation method for lattice polymers*, *Comp. Phys. Commun.* **61**, 309 (1990); *Erratum notice*, *ibid.* **71**, 343 (1992).
46. W. J. C. Orr, *Trans. Faraday Soc.* **43**, 12 (1947); E. W. Montroll, *Markoff Chains and Excluded Volume Effects in Polymer Chains*, *J. Chem. Phys.* **18**, 734 (1950).
47. P. Grassberger, *Pruned-enriched Rosenbluth method: Simulations of  $\theta$  polymers of chain length up to 1 000 000*, *Phys. Rev. E* **56**, 3682 (1997).
48. P. H. Verdier and W. H. Stockmayer, *Monte Carlo Calculations on the Dynamics of Polymers in Dilute Solution*, *J. Chem. Phys.* **36**, 227 (1962).
49. N. Madras and A. D. Sokal, *Nonergodicity of local, length-conserving Monte Carlo*

- algorithms for the self-avoiding walk*, J. Stat. Phys. **47**, 573 (1987).
50. S. Caracciolo, M. S. Causo, G. Ferraro, M. Papinutto, and A. Pelissetto, *Bilocal Dynamics for Self-Avoiding Walks*. J. Stat. Phys. **100**, 1111 (2000).
  51. S. Caracciolo, M. Papinutto, and A. Pelissetto, *Dynamic critical behavior of an extended reptation dynamics for self-avoiding walks*. Phys. Rev. E **65**, 031106 (2002).
  52. I. Carmesin and K. Kremer, *The Bond Fluctuation Method: A New Effective Algorithm for the Dynamics of Polymers in All Spatial Dimensions*, Macromolecules **21**, 2819 (1988).
  53. I. Carmesin and K. Kremer, *Static and dynamic properties of two-dimensional polymer melts*, J. Physique **51**, 915 (1990).
  54. H.-P. Deutsch and K. Binder, *Interdiffusion and self-diffusion in polymer mixtures: a Monte Carlo study*, J. Chem. Phys. **94**, 2294 (1991).
  55. B. Lobe and J. Baschnagel, *Monte Carlo simulation of the glass transition in two- and three-dimensional polymer melts: Influence of the spatial dimension*, J. Chem. Phys. **101**, 1616 (1994).
  56. A. Z. Panagiotopoulos, *On the equivalence of continuum and lattice models for fluids*, J. Chem. Phys. **112**, 7132 (2000).
  57. H.-P. Deutsch and R. Dickman, *Equation of state for athermal lattice chains in a 3d fluctuating bond model*, J. Chem. Phys. **93**, 8983 (1990).
  58. M. Wittkop, S. Kreitmeier, and D. Göritz, *Monte Carlo simulations of a single polymer chain under extension above and below the  $\Theta$  temperature*, Phys. Rev. E **53**, 838 (1996).
  59. J.-P. Hansen and I. R. McDonald, *Theory of Simple Liquids* (Academic Press, London, 1986).
  60. N. B. Wilding, M. Müller, and K. Binder, *Chain length dependence of the polymer-solvent critical point parameters*, J. Chem. Phys. **105**, 802 (1996).
  61. G. S. Grest and K. Kremer, *Molecular dynamics simulation of polymers in the presence of a heat bath*, Phys. Rev. A **33**, 3628 (1986); K. Kremer and G. S. Grest, *Dynamics of entangled linear polymer melts: A molecular-dynamics simulation*, J. Chem. Phys. **92**, 5057 (1990).
  62. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, Oxford, 1994).
  63. B. Smit, *Phase diagram of Lennard-Jones fluids*, J. Chem. Phys. **96**, 8639 (1992).
  64. C. W. Yong, J. H. R. Clarke, J. J. Freire, and M. Bishop, *The theta condition for linear polymer chains in continuous space and three dimensions*, J. Chem. Phys. **105**, 9666 (1996).
  65. C. Bennemann, W. Paul, K. Binder, and B. Dünweg, *Molecular-dynamics simulations of the thermal glass transition in polymer melts:  $\alpha$ -relaxation behavior*, Phys. Rev. E **57**, 843 (1998).
  66. M. Müller and L. Gonzalez MacDowell, *Interface and Surface Properties of Short Polymers in Solution: Monte Carlo Simulations and Self-Consistent Field Theory*, Macromolecules **33**, 3902 (2000).
  67. H. Meyer, O. Biermann, R. Faller, D. Reith, and F. Müller-Plathe, *Coarse graining of nonbonded inter-particle potentials using automatic simplex optimization to fit structural properties*, J. Chem. Phys. **113**, 6264 (2000).
  68. G. S. Grest and M. Murat, *Computer Simulations of Tethered Chains*. In: *Monte Carlo*

- and *Molecular Dynamics Simulations in Polymer Science* (Oxford University Press, New York, 1995), pp. 476–578.
69. C. F. Abrams and K. Kremer, *The effect of bond length on the structure of dense bead-spring polymer melts*, J. Chem. Phys. **115**, 2776 (2001); *Effects of excluded volume and bond length on the dynamics of dense bead-spring polymer melts*, *ibid.* **116**, 3162 (2002).
  70. L. Mattioni *et al.*, in preparation.
  71. H. E. A. Huitema and J. P. van der Eerden, *Can Monte Carlo simulation describe dynamics? A test on Lennard-Jones systems*, J. Chem. Phys. **110**, 3267 (1999).
  72. W. Paul, *Anomalous diffusion in polymer melts*, Chem. Phys. **284**, 59 (2001).
  73. See e.g. <http://hostingworks.com/support/dict.phtml?foldoc=complexity>.
  74. F. T. Wall and F. Mandel, *Macromolecular dimensions obtained by an efficient Monte Carlo method without sample attrition*, J. Chem. Phys. **63**, 4592 (1975); F. T. Mandel, *Macromolecular dimensions obtained by an efficient Monte Carlo method: The mean square end-to-end separation*, J. Chem. Phys. **70**, 3984 (1979).
  75. E. Leontidis, B. M. Forrest, A. H. Widmann, and U. W. Suter, *Monte Carlo Algorithms for the Atomistic Simulation of Condensed Polymer Phases*, Faraday Trans. **91**, 2355 (1995).
  76. J. Reiter, *Monte Carlo Simulations of Linear and Cyclic Chains on Cubic and Quadratic Lattices*, Macromolecules **23**, 3811 (1990).
  77. J. Reiter and E. R. Duering, *Monte Carlo simulation of freely jointed chains with variable bond lengths*, Macromol. Theory Simul. **4**, 667 (1995).
  78. J. Reiter, T. Edling, and T. Pakula, *Monte Carlo simulation of lattice models for macromolecules at high densities*, J. Chem. Phys. **93**, 837 (1990).
  79. J. M. Deutsch, *The dynamics of entangled polymers*, J. Physique **48**, 141 (1987).
  80. L. Mattioni, J. P. Wittmer, J. Baschnagel, J.-L. Barrat, E. Luijten, *Dynamical Properties of the Slithering Snake Algorithm: A numerical test of the activated reptation hypothesis*, Eur. Phys. J. E **10**, 369 (2003).
  81. H. Weber, *Monte-Carlo-Simulationen der Gasdiffusion in Polymermatrizen*, Dissertation (Universität Mainz, 1997), unpublished.
  82. A. N. Semenov, *The Effect of Polymer Density Fluctuations on the Dynamics of High Molecular-Weight Polymers*, In: *Theoretical Challenges in the Dynamics of Complex Fluids* (Kluwer, Dordrecht, 1997), pp. 77–86.
  83. A. N. Semenov and M. Rubinstein, *Dynamics of strongly entangled polymer systems: activated reptation*, Eur. Phys. J. B **1**, 47 (1998).
  84. M. Wolfgang, J. Baschnagel, and K. Binder, *Monte-Carlo Simulation of 3-Dimensional Glassy Polymer Melts: Reptation versus Single Monomer Dynamics*, J. Phys. II **5**, 1035 (1995).
  85. V. Tries, W. Paul, J. Baschnagel, and K. Binder, *Modeling polyethylene with the bond fluctuation model*, J. Chem. Phys. **106**, 738 (1997).
  86. S. W. Sides, J. Curro, G. S. Grest, M. J. Stevens, T. Soddemann, A. Habenschuss, and J. D. Londono, *Structure of Poly(dimethylsiloxane) Melts: Theory, Simulation, and Experiment*, Macromolecules **35**, 6455 (2002); J. G. Curro, J. D. Weinhold, J. J. Rajasekaran, A. Habenschuss, J. D. Londono, and J. D. Honeycutt, *Intermolecular Packing in Stereoregular Polypropylene Liquids: Comparison between Theory and*

- X-ray Scattering Experiments*, *Macromolecules* **30**, 6264 (1997).
87. J. G. Curro, E. B. Webb III, G. S. Grest, J. D. Weinhold, M. Pütz, and J. D. McCoy, *Comparisons between integral equation theory and molecular dynamics simulations for realistic models of polyethylene liquids*, *J. Chem. Phys.* **111**, 9073 (1999).
  88. W. Janke, *Monte Carlo Simulations of Spin Systems*. In: *Computational Physics* (Springer, Berlin-Heidelberg, 1996), pp. 11–43.
  89. N. Madras and A. D. Sokal, *The pivot algorithm: a highly efficient Monte Carlo method for the self-avoiding walk*, *J. Stat. Phys.* **50**, 109 (1988).
  90. T. Kennedy, *A Faster Implementation of the Pivot Algorithm for Self-Avoiding Walks*, *J. Stat. Phys.* **106**, 407 (2002).
  91. B. Li, N. Madras, and A. D. Sokal, *Critical Exponents, Hyperscaling, and Universal Amplitude Ratios for Two- and Three-dimensional Self-Avoiding Walks*, *J. Stat. Phys.* **80**, 661 (1995).
  92. P. Grassberger, P. Sutter, and L. Schäfer, *Field theoretic and Monte Carlo analysis of the Domb-Joyce model*, *J. Phys. A* **30**, 7039 (1997).
  93. R. Auhl, R. Everaers, G. S. Grest, K. Kremer, and S. J. Plimpton, *Equilibration of Long Chain Polymer Melts in Computer Simulations*, preprint (cond-mat/0306026v1).
  94. N. Ch. Karayiannis, A. E. Giannousaki, V. G. Mavrantzas, and D. N. Theodorou, *Atomistic Monte Carlo simulation of strictly monodisperse long polyethylene melts through a generalized chain bridging algorithm*, *J. Chem. Phys.* **117**, 5465 (2002); N. Ch. Karayiannis, V. G. Mavrantzas, and D. N. Theodorou, *A Novel Monte Carlo Scheme for the Rapid Equilibration of Atomistic Model Polymer Systems of Precisely Defined Molecular Architecture*, *Phys. Rev. Lett.* **88**, 105503 (2002).
  95. J. Batoulis and K. Kremer, *Statistical properties of biased sampling methods for long polymer chains*, *J. Phys. A: Math. Gen.* **21**, 127 (1988).
  96. M. N. Rosenbluth and A. W. Rosenbluth, *Monte Carlo simulations of the average extension of molecular chains*, *J. Chem. Phys.* **23**, 356 (1955).
  97. C. M. Guttman, *Relationship of the Unweighted Rosenbluth and Rosenbluth Walk to a Polymer Chain at the  $\Theta$  Point*, *Macromolecules* **19**, 833 (1986).
  98. M. Wittkop, S. Kreitmeier, and D. Göritz, *The collapse transition of a single polymer chain in two and three dimensions: A Monte Carlo study*, *J. Chem. Phys.* **104**, 3373 (1996).
  99. A. M. Rubio, J. J. Freire, C. W. Yong, and J. H. R. Clarke, *Conformation properties of polymer chains in the theta region*, *J. Chem. Phys.* **111**, 1302 (1999).
  100. P. Grassberger, *Go with the winners: a general Monte Carlo strategy*, *Comp. Phys. Comm.* **147**, 64 (2002).
  101. J. Hager and L. Schäfer,  *$\Theta$ -point behavior of diluted polymer solutions: Can one observe the universal logarithmic corrections predicted by field theory?*, *Phys. Rev. E* **60**, 2071 (1999).
  102. M. S. Causo, B. Coluzzi, and P. Grassberger, *Simple model for the DNA denaturation transition*, *Phys. Rev. E* **62**, 3958 (2000).
  103. H. Frauenkron and P. Grassberger, *Critical unmixing of polymer solutions*, *J. Chem. Phys.* **107**, 9599 (1997).
  104. S. Consta, N. B. Wilding, D. Frenkel, and Z. Alexandrowicz, *Recoil growth: An efficient simulation method for multi-polymer systems*, *J. Chem. Phys.* **110**, 3220 (1999).
  105. S. Consta, T. J. H. Vlugt, J. W. Hoeth, B. Smit, and D. Frenkel, *Recoil growth algo-*



- rithm for chain molecules with continuous interactions*, Mol. Phys. **97**, 1243 (1999).
106. J. J. de Pablo, Q. Yan, and F. A. Escobedo, *Simulation of Phase Transitions in Fluids*, Annu. Rev. Phys. Chem. **50**, 377 (1999).
  107. M. Müller, K. Binder, and E. V. Albano, *Phase equilibria in thin polymer films*, Int. J. Mod. Phys. B **15**, 1867 (2001).
  108. See e.g. H. Meirovitch, *On the simulation of the entropy of macromolecules with different flexibilities*, J. Chem. Phys. **114**, 3859 (2001); *Entropy, pressure, and chemical potential of multiple chain systems from computer simulation. I. Application of the scanning method*, ibid. **97**, 5803 (1992).
  109. M. Aichele *et al.*, in preparation; M. Aichele, *Simulation Studies of Correlation Functions and Relaxation in Polymeric Systems*, Dissertation (Universität Mainz, 2003), unpublished.
  110. P. C. Hemmer and S. Hemmer, *Trapping of genuine self-avoiding walks*, Phys. Rev. A **34**, 3304 (1986).
  111. S. Caracciolo, A. Pelissetto, and A. D. Sokal, *Universal distance ratios for two-dimensional self-avoiding walks: corrected conformal-invariance predictions*, J. Phys. A: Math. Gen. **23**, L969 (1990).
  112. M. Murat and K. Kremer, *From many monomers to many polymers: Soft ellipsoid model for polymer melts and mixtures*, J. Chem. Phys. **108**, 4340 (1998).
  113. F. Eurich and P. Maass, *Soft ellipsoid model for Gaussian polymer chains*, J. Chem. Phys. **114**, 7655 (2001).
  114. A. A. Louis, P. G. Bolhuis, R. Finken, V. Krakoviack, E. J. Meijer, and J. P. Hansen, *Coarse-graining polymers as soft colloids*, Physica A **306**, 251 (2002).
  115. B. A. Berg, *Generalized ensemble simulations for complex systems*, Comp. Phys. Comm. **147**, 52 (2002).
  116. A. D. Bruce and N. B. Wilding, *Computational strategies for mapping equilibrium phase diagrams*, preprint (cond-mat/0210457v1).
  117. N. B. Wilding and M. Müller, *Accurate measurements of the chemical potential of polymer systems by Monte Carlo simulation*, J. Chem. Phys. **101**, 4324 (1994).
  118. A. Bunker and B. Dünweg, *Parallel excluded volume tempering for polymer melts*, Phys. Rev. E **63**, 016701 (2000).
  119. Q. Yan and J. J. de Pablo, *Hyperparallel tempering Monte Carlo simulation of polymeric systems*, J. Chem. Phys. **113**, 1276 (2000).
  120. F. Müller-Plathe, *Scale Hopping in Computer Simulations of Polymers*, Soft Materials **1**, 1 (2003).
  121. G. Galli and M. Pasquarello, *First-Principles Molecular Dynamics*. In: *Computer Simulation in Chemical Physics* (Dordrecht, Kluwer, 1993), pp. 261–313; D. Marx and J. Hutter, *Ab initio Molecular Dynamics: Theory and Implementation*. In: *Modern Methods and Algorithms of Quantum Chemistry* (NIC Series Vol. **3**, 2000), pp. 329–477 (available from <http://www.fz-juelich.de/nic-series>).
  122. L. Delle Site, C. F. Abrams, A. Alavi, and K. Kremer, *Polymers near metal surfaces: Selective adsorption and global conformations*, Phys. Rev. Lett. **89**, 156103 (2002).
  123. D. N. Theodorou and U. W. Suter, *Detailed Molecular Structure of a Vinyl Polymer Glass*, Macromolecules **18**, 1467 (1985).
  124. T. A. Weber and E. Helfand, *Molecular dynamics simulation of polymers. I. Structure*, J. Chem. Phys. **71**, 4760 (1979).

125. R. L. McGreevy and L. Pusztai, *Reverse Monte Carlo simulation: A new technique for the determination of disordered structure*, Mol. Sim. **1**, 359 (1988).
126. A. K. Soper, *Empirical Monte Carlo simulation of fluid structure*, Chem. Phys. **202**, 295 (1996); *The quest for the structure of water and aqueous solutions*, J. Phys.: Condens. Matter **9**, 2717 (1997).
127. D. Reith, H. Meyer, and F. Müller-Plathe, *Mapping atomistic to coarse-grained polymer models using automatic simplex optimization to fit structural properties*, Macromolecules **34**, 2335 (2001).
128. J. T. Padding and W. J. Briels, *Time and length scales of polymer melts studied by coarse-grained molecular dynamics simulations*, J. Chem. Phys. **117**, 925 (2002).
129. H. Meyer and F. Müller-Plathe, *Formation of Chain-Folded Structures in Supercooled Polymer Melts Examined by MD Simulations*, Macromolecules **35**, 1241 (2002); H. Meyer, *Structure Formation and Chain-Folding in Supercooled Polymer Melts. Some Ideas from MD Simulations with a Coarse-Grained Model*, chapter 10 of Ref. 25.